CMPSCI 670: Computer Vision

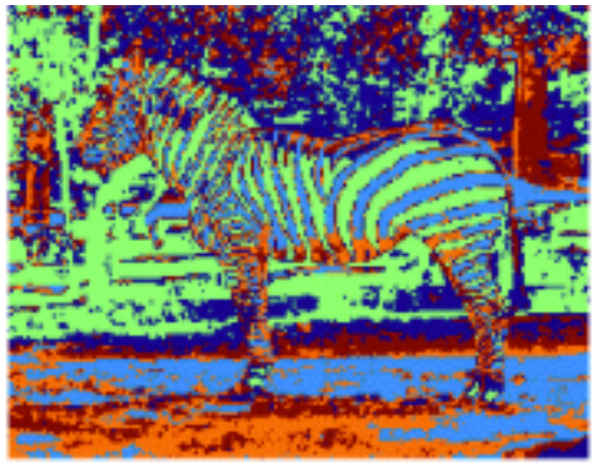
Alignment continued …

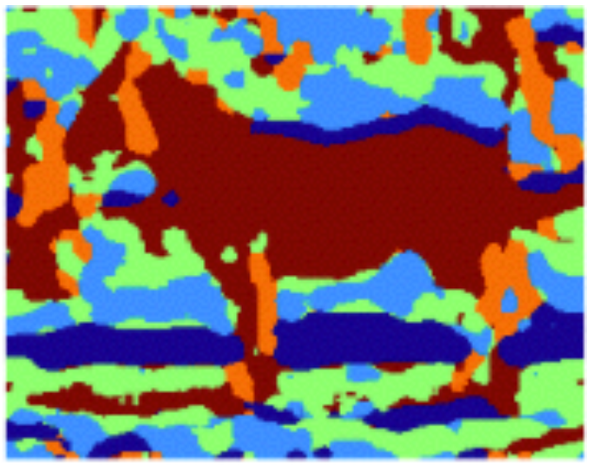
University of Massachusetts, Amherst
October 22, 2014

Instructor: Subhransu Maji

Administrivia

- Homework 4 posted (due Nov 5)
  - Grouping and texture





- Office hours ~~today~~ tomorrow 2:30 - 3:30 pm

Distinguished Lecturer Series

Jennifer Chayes
Microsoft Research

Wednesday, October 22, 2014
4:00pm - 5:00pm
Computer Science Building, Room 151
Faculty Host: Andrew McGregor

"Age of Networks"

Everywhere we turn, we find that networks can be used to describe
the World Wide Web, and a variety of online social networks. In ec
negative effects of a global networked economy. In epidemiology,
complicated by mutation of the disease agents. In biomedical resea

Alignment so far …

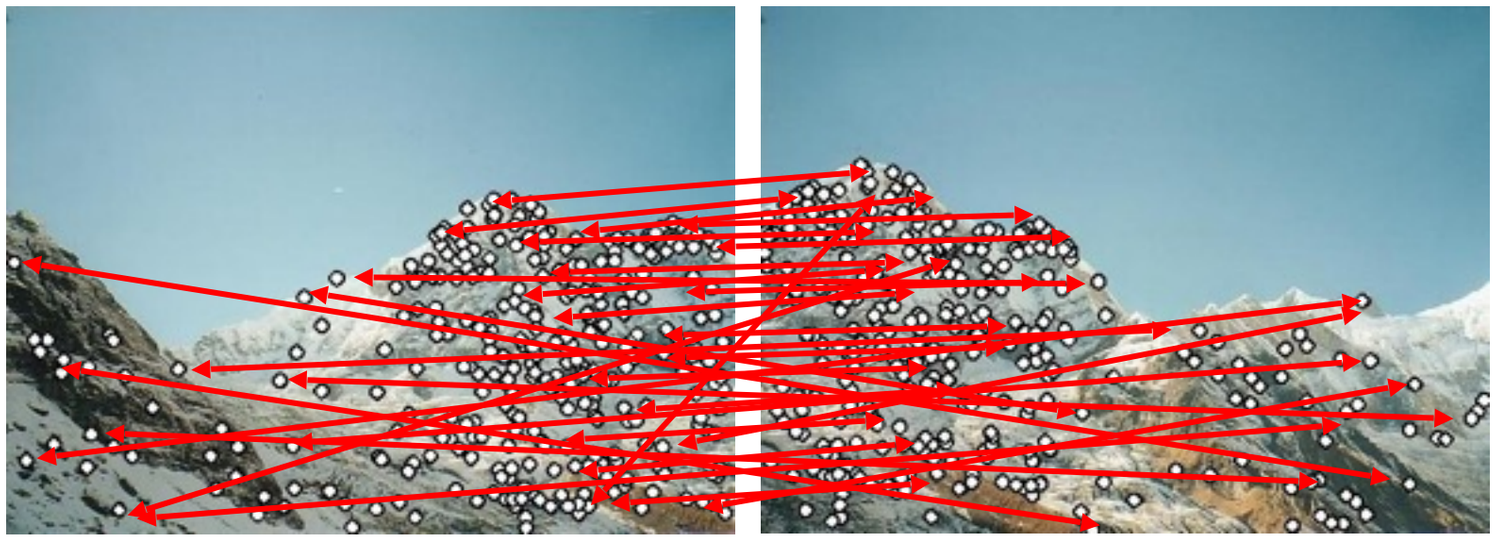
- What are the alignment problems in computer vision?
  - rigid vs. deformable (non-rigid)

- Good features to match
  - Invariance properties
  - Local features: the SIFT descriptor

- Alignment algorithms
  - Rigid alignment: RANSAC
    - Application: panoramic photo stitching
  - Non-rigid alignment
    - Application: shape matching

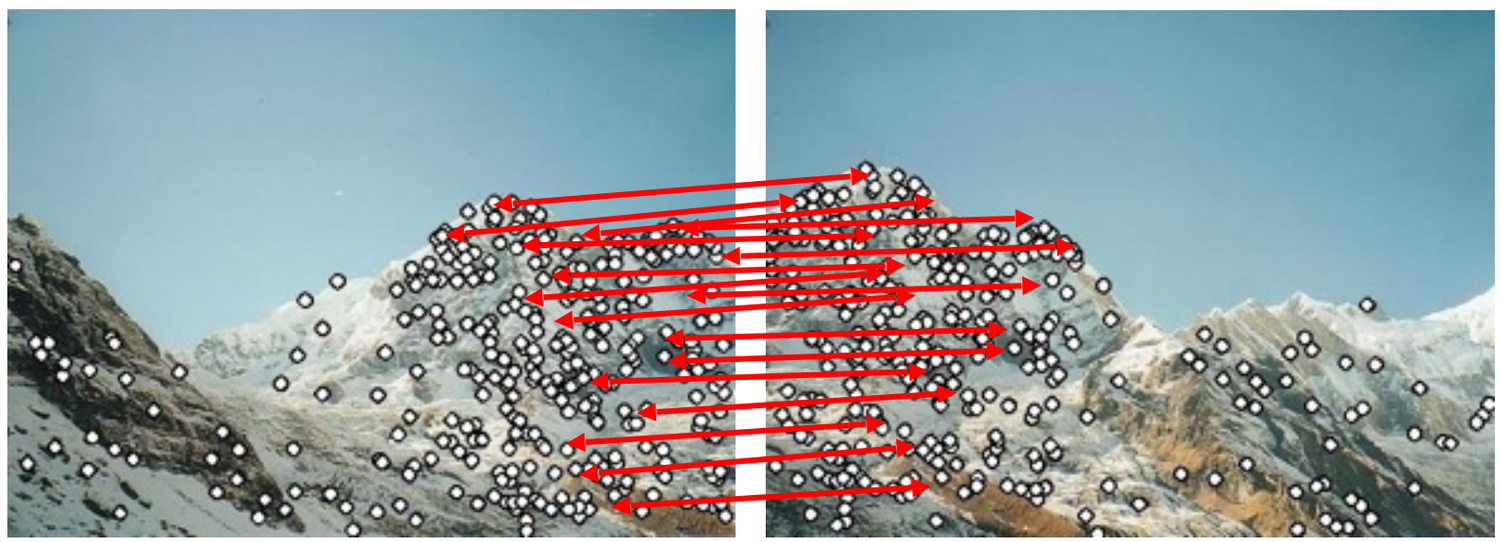
A framework for alignment

- Matching local features
  - Local information used, can contain outliers
  - But hopefully enough of these matches are good

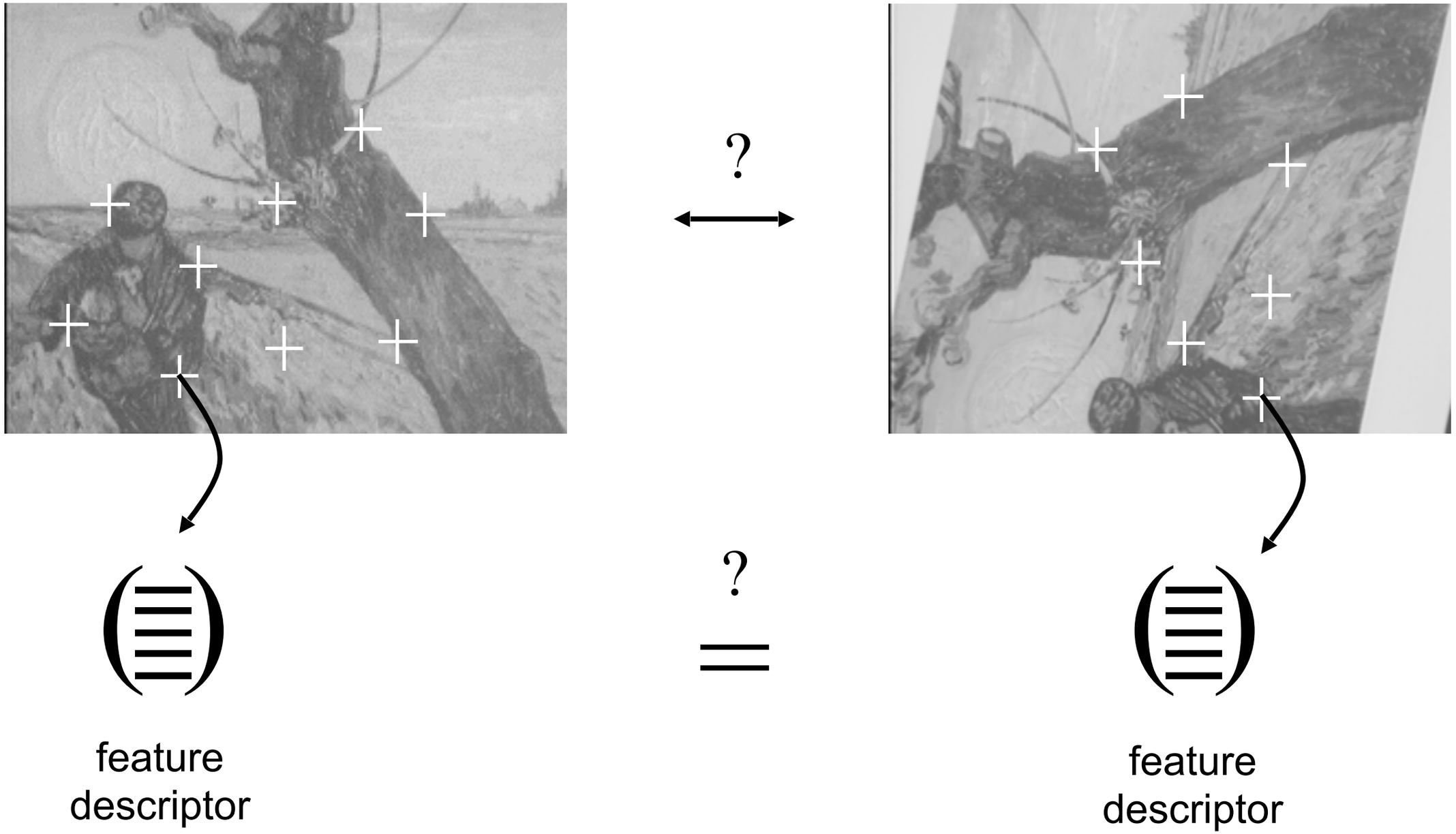


A framework for alignment

- Matching local features
  - Local information used, can contain outliers
  - But hopefully enough of these matches are good
- Consensus building
  - Aggregate the good matches and find a transformation that explains these matches



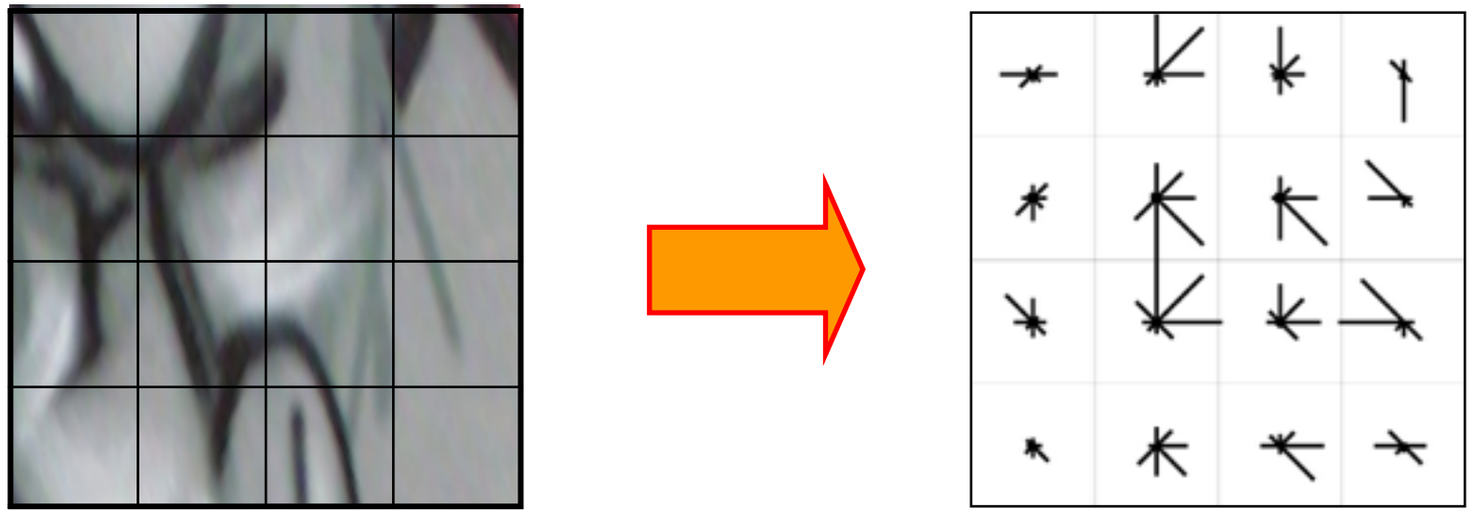
Generating putative correspondences



- Need to compare *feature descriptors* of local patches surrounding interest points

Feature descriptors: SIFT

- Descriptor computation:
  - Divide patch into 4x4 sub-patches
  - Compute histogram of gradient orientations (8 reference angles) inside each sub-patch
  - Resulting descriptor: 4x4x8 = 128 dimensions



David G. Lowe. "Distinctive image features from scale-invariant keypoints." *IJCV* 60 (2), pp. 91-110, 2004.

Feature descriptors: SIFT

- Descriptor computation:
  - Divide patch into 4x4 sub-patches
  - Compute histogram of gradient orientations (8 reference angles) inside each sub-patch
  - Resulting descriptor: 4x4x8 = 128 dimensions

- Advantage over raw vectors of pixel values
  - Gradients less sensitive to illumination change
  - Pooling of gradients over the sub-patches achieves robustness to small shifts, but still preserves some spatial information

David G. Lowe. "Distinctive image features from scale-invariant keypoints." *IJCV* 60 (2), pp. 91-110, 2004.

Alignment

- What are the alignment problems in computer vision?
  - rigid vs. deformable (non-rigid)

- Good features to match
  - Invariance properties
  - Local features: the SIFT descriptor

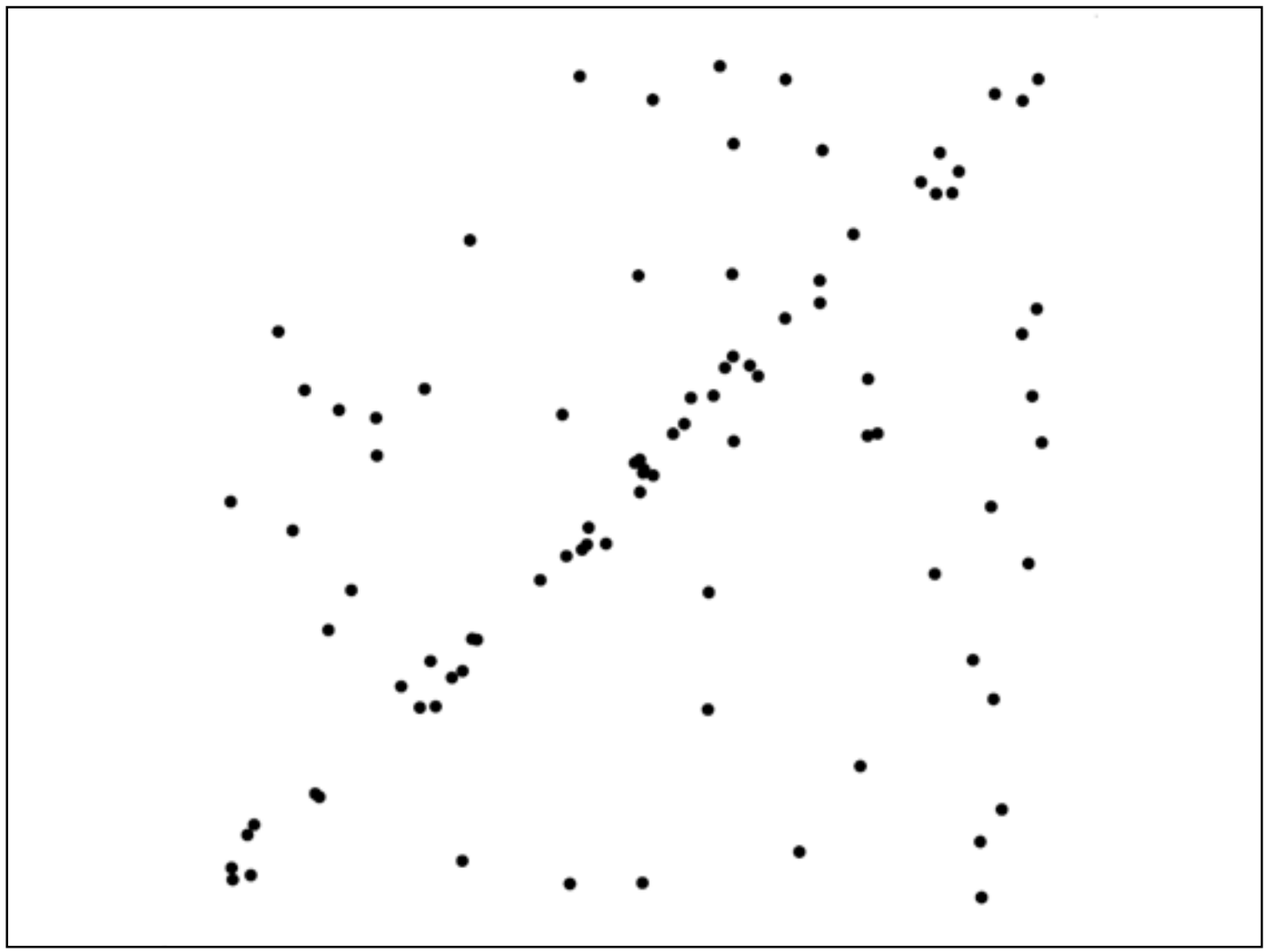
- Alignment algorithms
  - Rigid alignment: RANSAC
    - Application: panoramic photo stitching
  - Non-rigid alignment
    - Application: shape matching

RANSAC

- **Ran**dom **Sa**mple **C**onsensus
  - Choose a small subset of points uniformly at random
  - Fit a model to that subset
  - Find all remaining points that are “close” to the model and reject the rest as outliers
  - Do this many times and choose the best model

M. A. Fischler, R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Comm. of the ACM, Vol 24, pp 381-395, 1981.

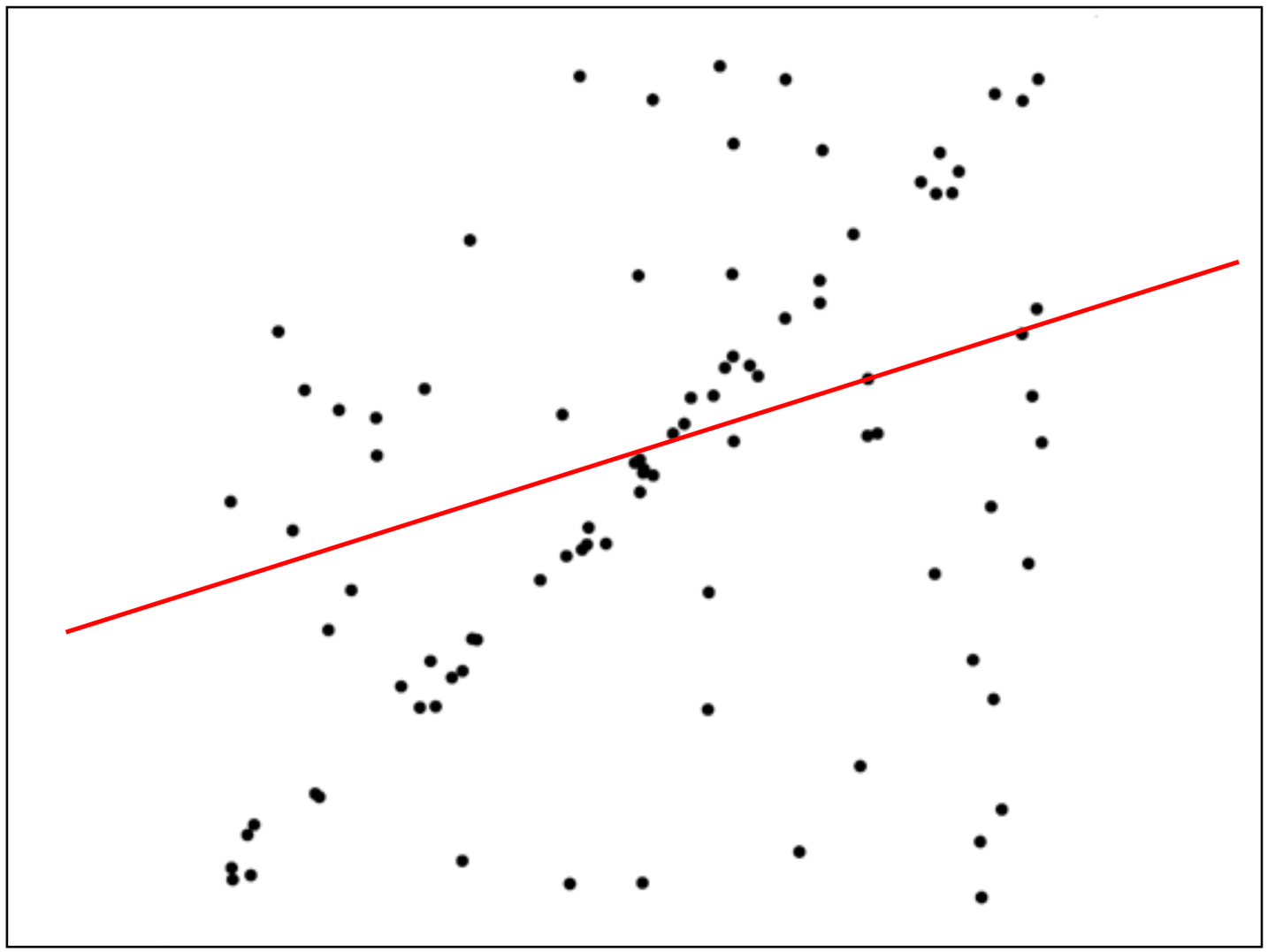
RANSAC for line fitting example



Source: R. Raguram

RANSAC for line fitting example

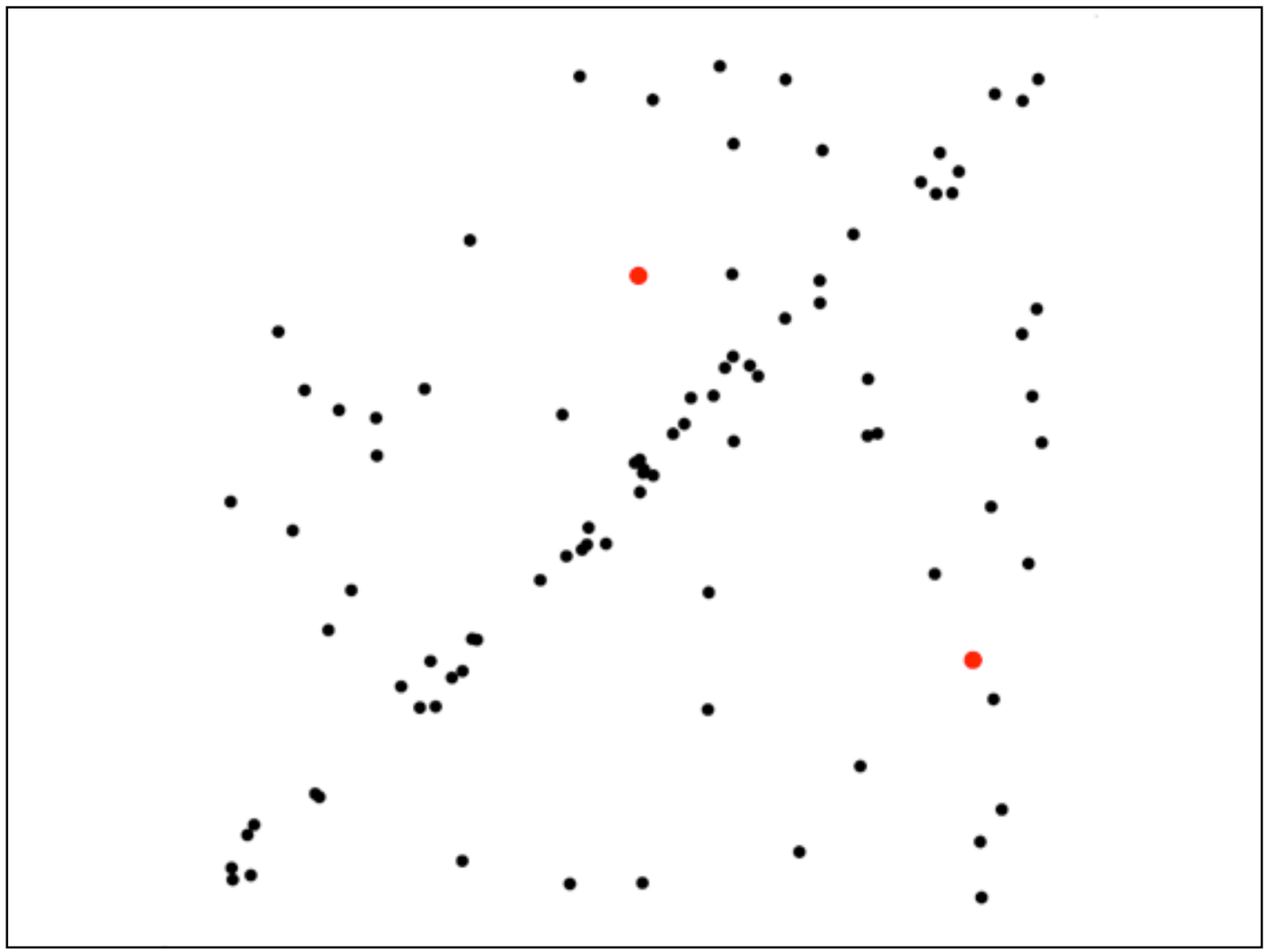
$$\min_{a,b} \sum_i (ax_i + b - y_i)^2$$



Least-squares fit

Source: R. Raguram

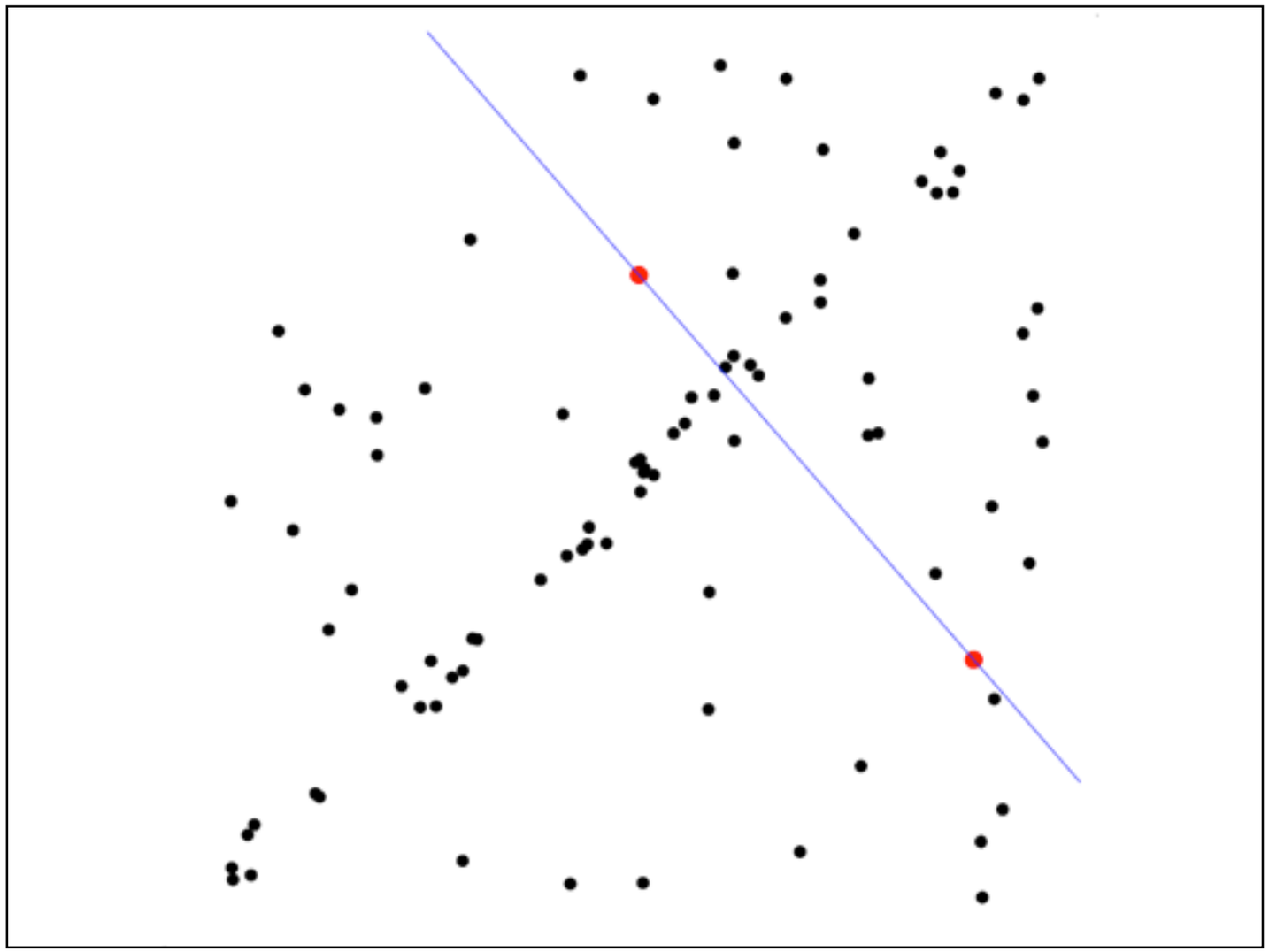
RANSAC for line fitting example



1. Randomly select minimal subset of points

Source: R. Raguram

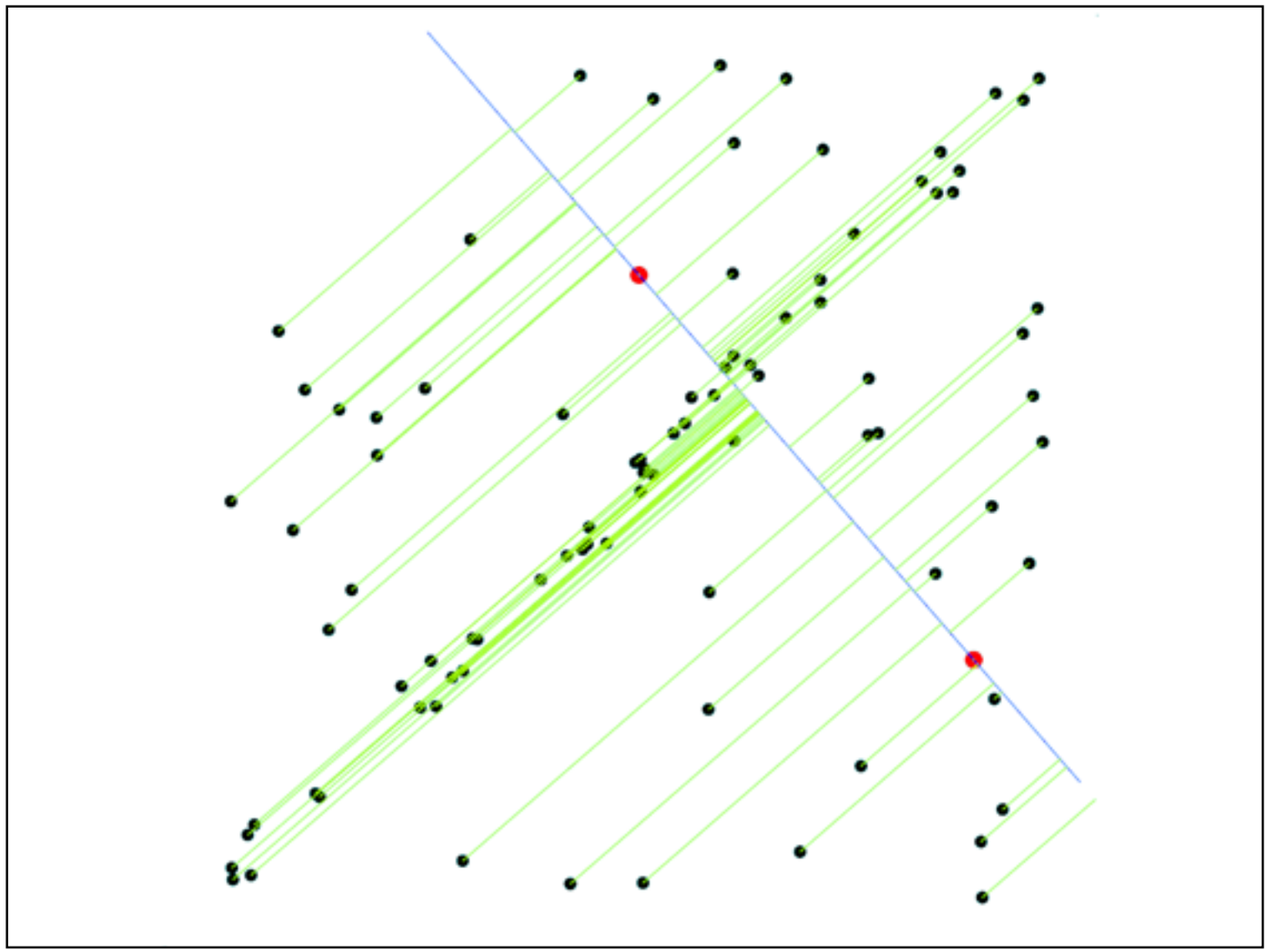
RANSAC for line fitting example



1. Randomly select minimal subset of points
2. Hypothesize a model

Source: R. Raguram

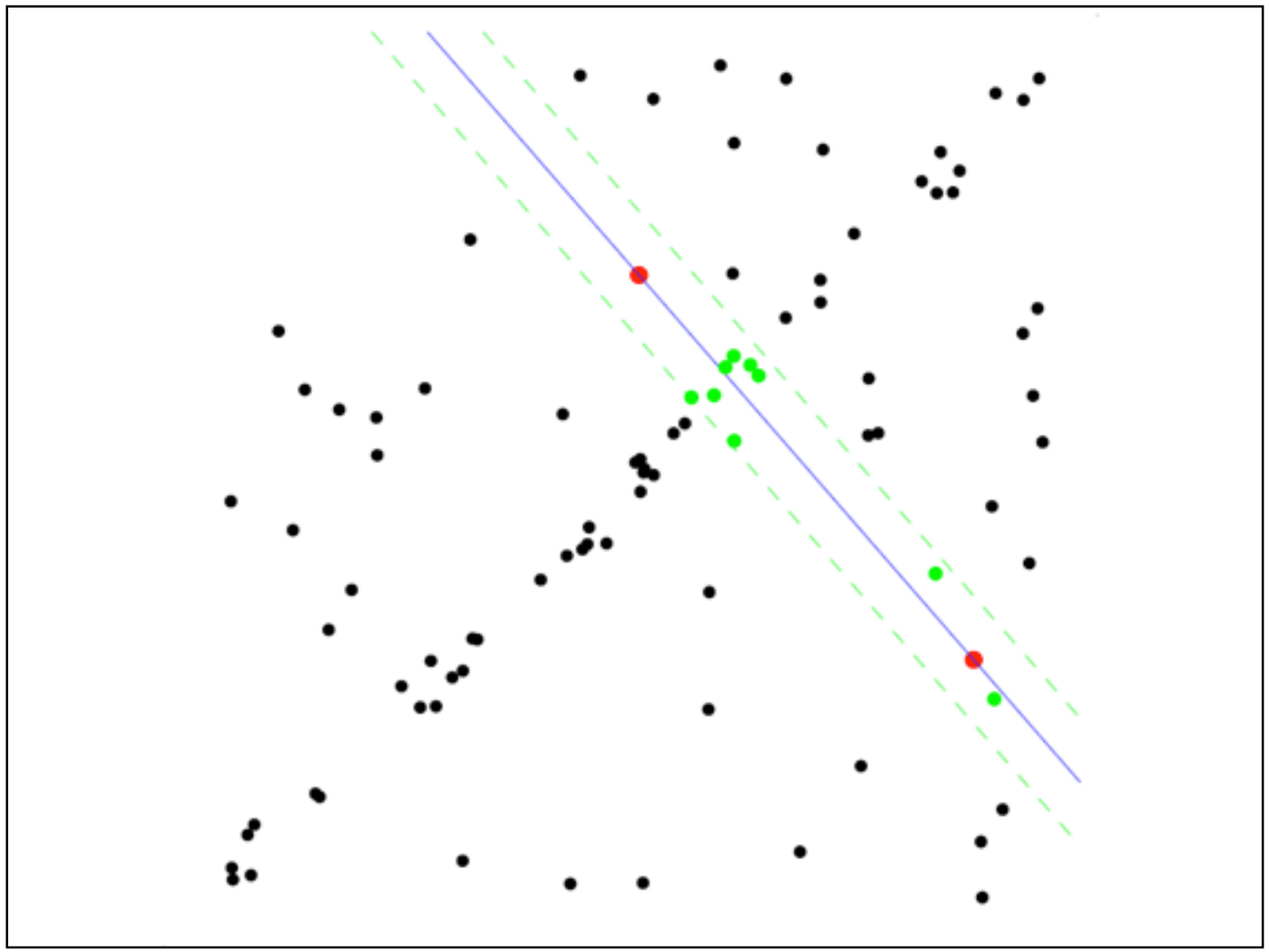
RANSAC for line fitting example



1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function

Source: R. Raguram

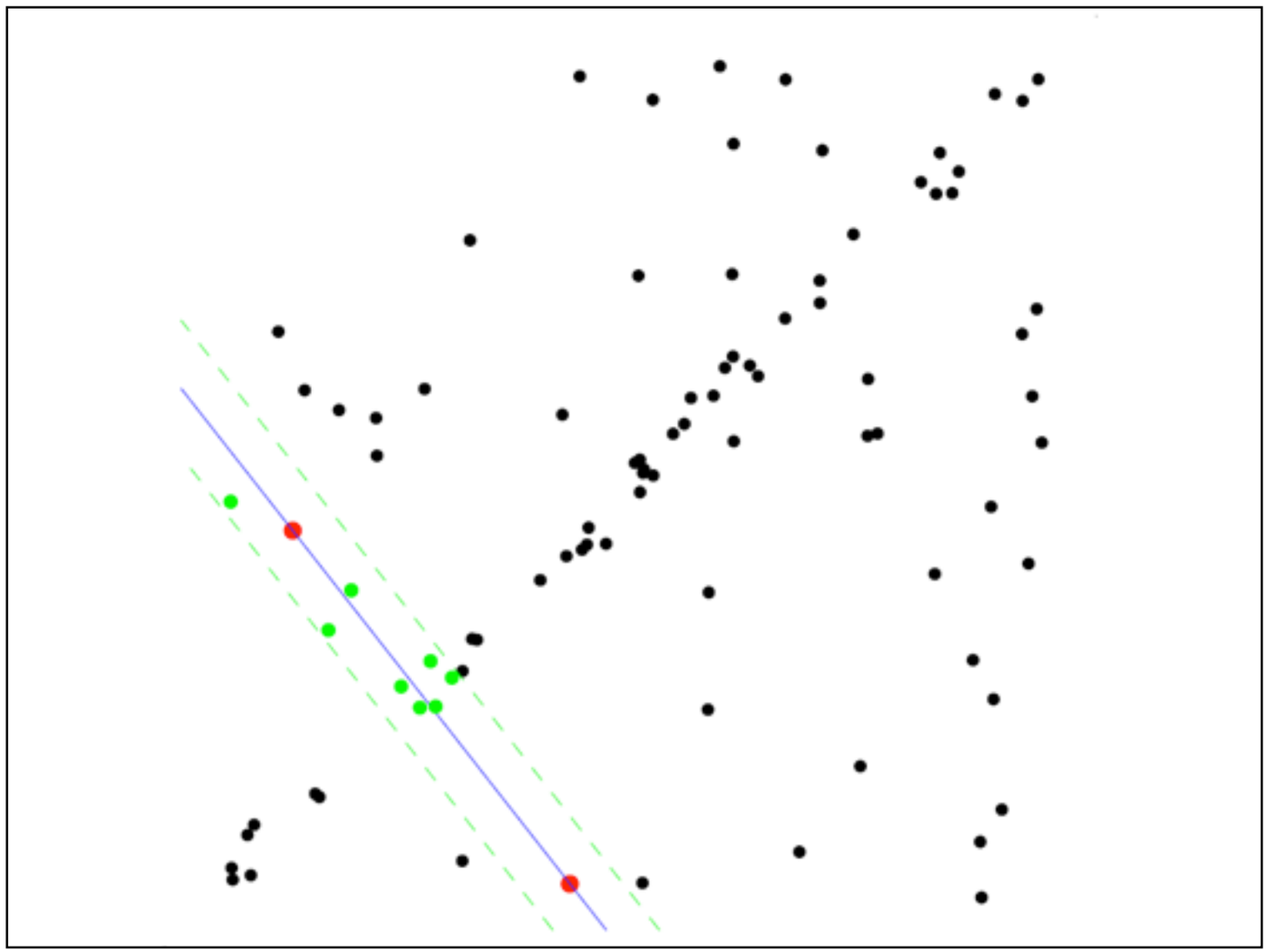
RANSAC for line fitting example



1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model

Source: R. Raguram

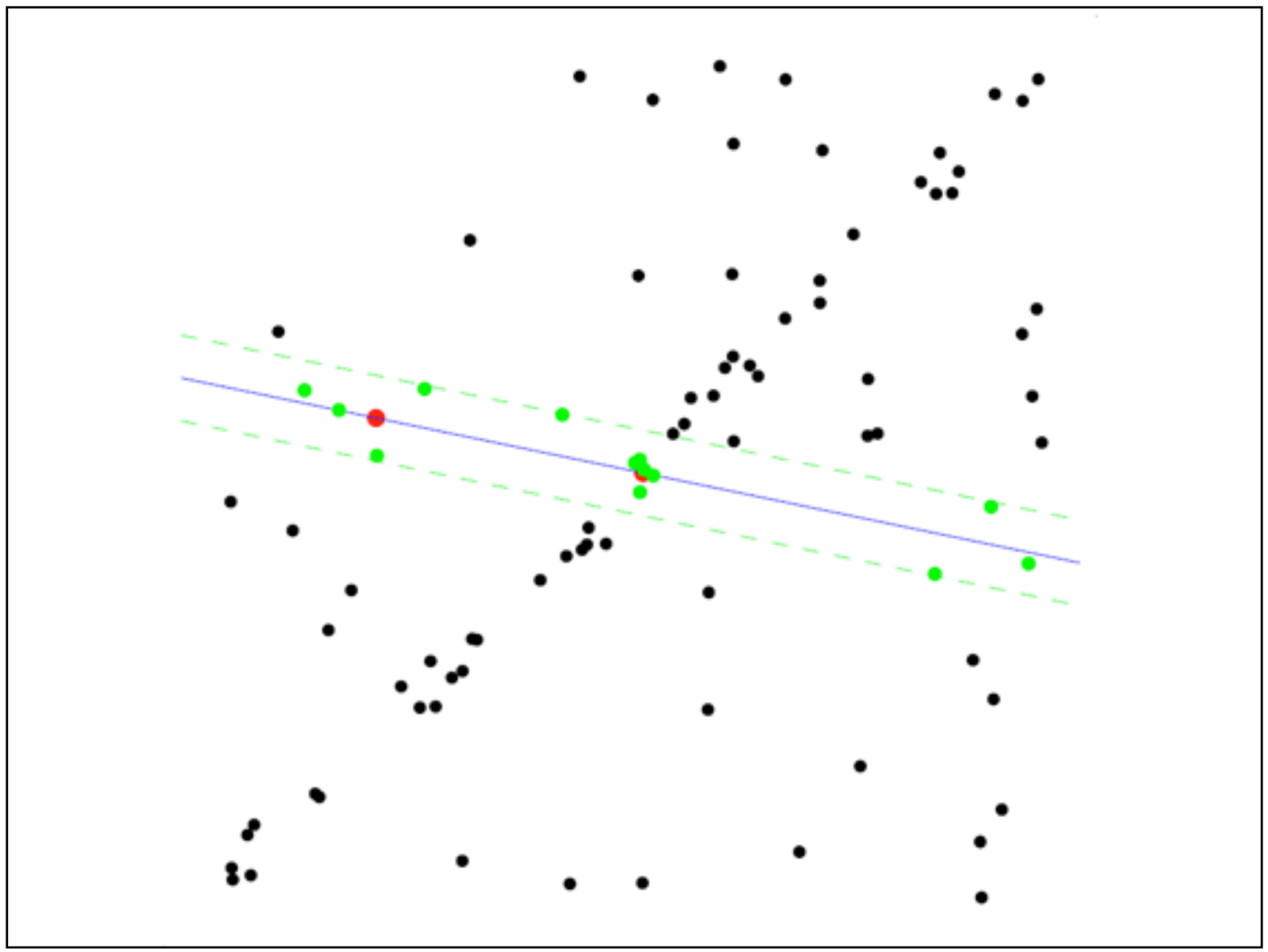
RANSAC for line fitting example



1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model
5. Repeat *hypothesize-and-verify* loop

Source: R. Raguram

RANSAC for line fitting example

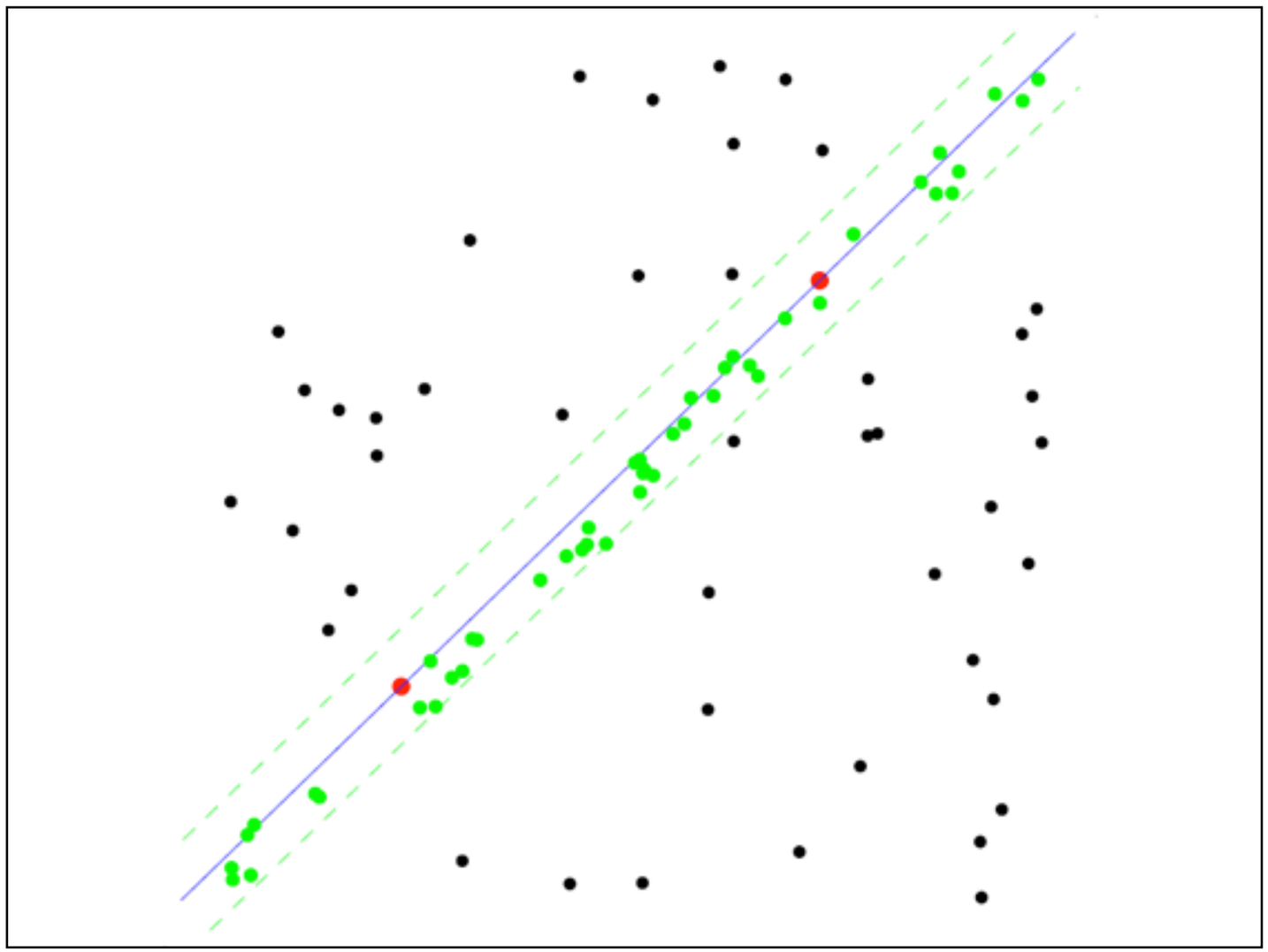


1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model
5. Repeat *hypothesize-and-verify* loop

Source: R. Raguram

RANSAC for line fitting example

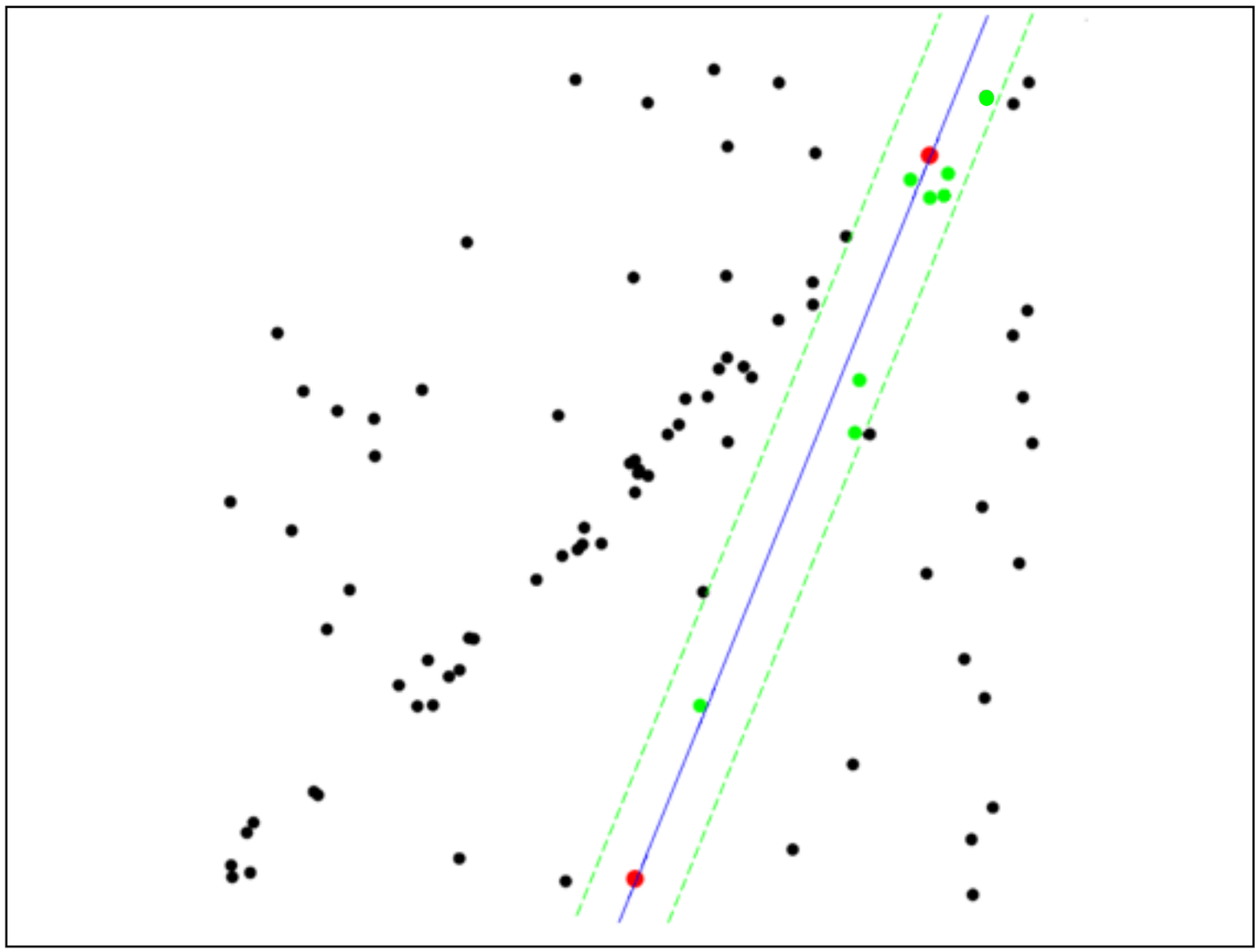
Uncontaminated sample



1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model
5. Repeat *hypothesize-and-verify* loop

Source: R. Raguram

RANSAC for line fitting example

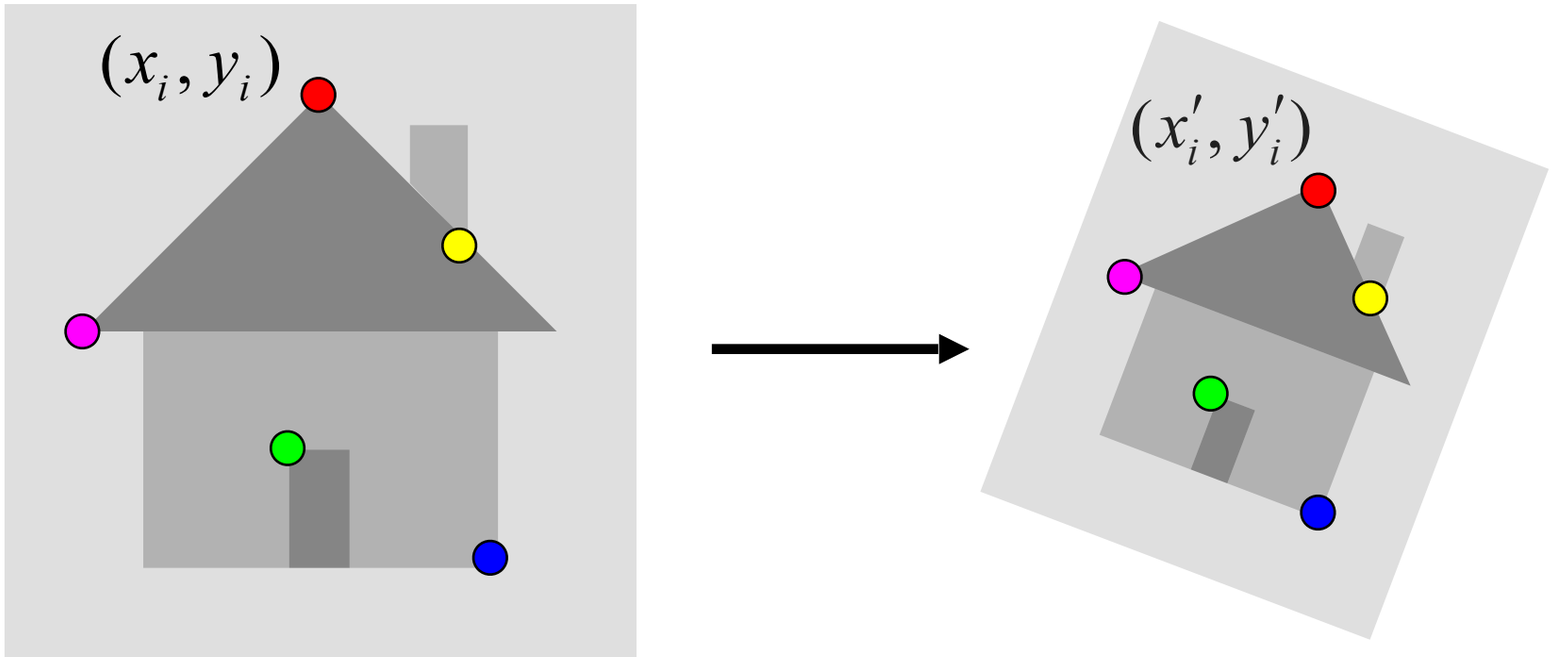


1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model
5. Repeat *hypothesize-and-verify* loop

Source: R. Raguram

Fitting an affine transformation

- Assume we know the correspondences, how do we get the transformation?



$$\begin{bmatrix} x_i' \\ y_i' \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}$$

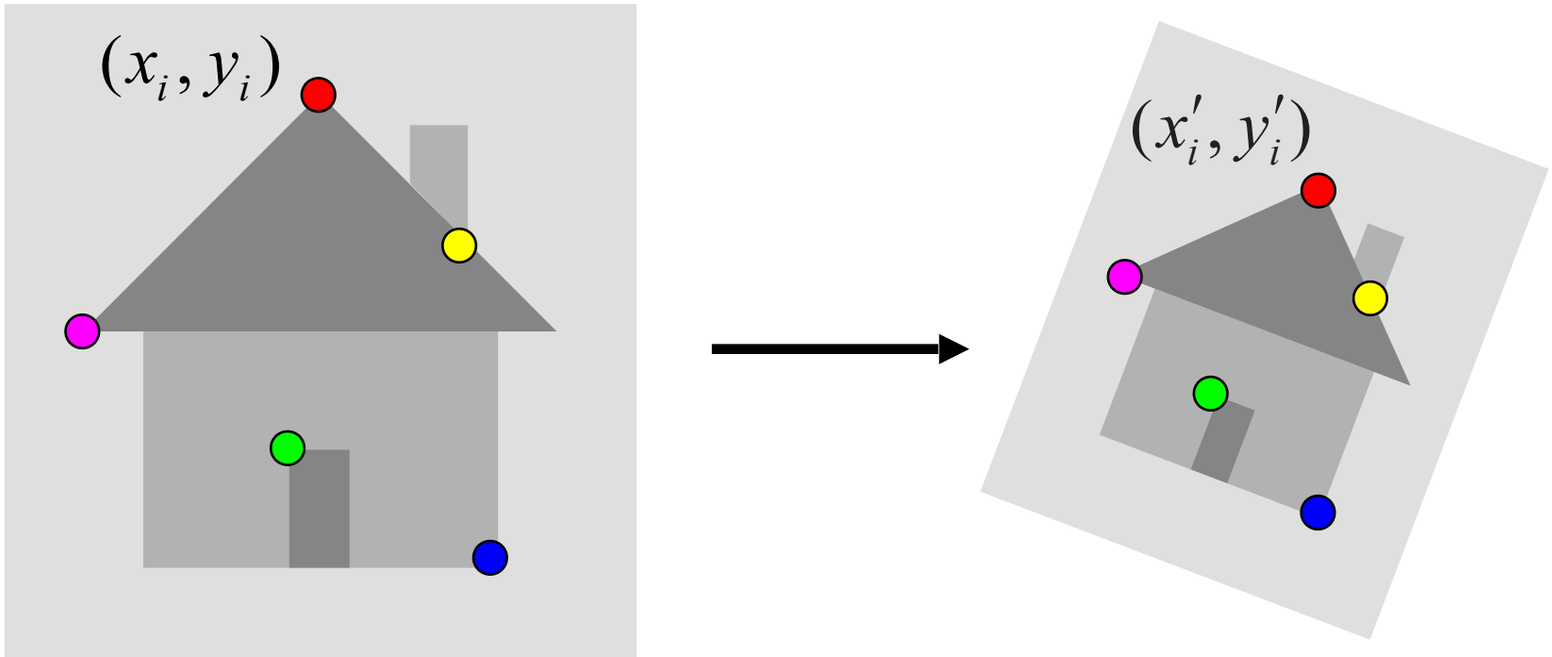
$$\mathbf{x}_i' = \mathbf{M}\mathbf{x}_i + \mathbf{t}$$

Want to find **M**, **t** to minimize

$$\sum_{i=1}^{n} \| \mathbf{x}_i' - \mathbf{M}\mathbf{x}_i - \mathbf{t} \|^2$$

Fitting an affine transformation

- Assume we know the correspondences, how do we get the transformation?



$$\begin{bmatrix} x_i' \\ y_i' \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}$$

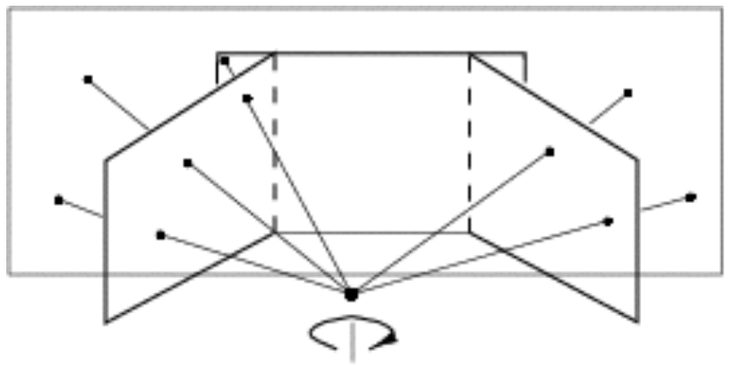
$$\begin{bmatrix} & & \cdots & & & \\ x_i & y_i & 0 & 0 & 1 & 0 \\ 0 & 0 & x_i & y_i & 0 & 1 \\ & & \cdots & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} \cdots \\ x_i' \\ y_i' \\ \cdots \end{bmatrix}$$

Fitting an affine transformation

$$\begin{bmatrix} & & \cdots & & & \\ x_i & y_i & 0 & 0 & 1 & 0 \\ 0 & 0 & x_i & y_i & 0 & 1 \\ & & \cdots & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} \cdots \\ x_i' \\ y_i' \\ \cdots \end{bmatrix}$$

- Linear system with six unknowns
- Each match gives us two linearly independent equations: need at least three to solve for the transformation parameters

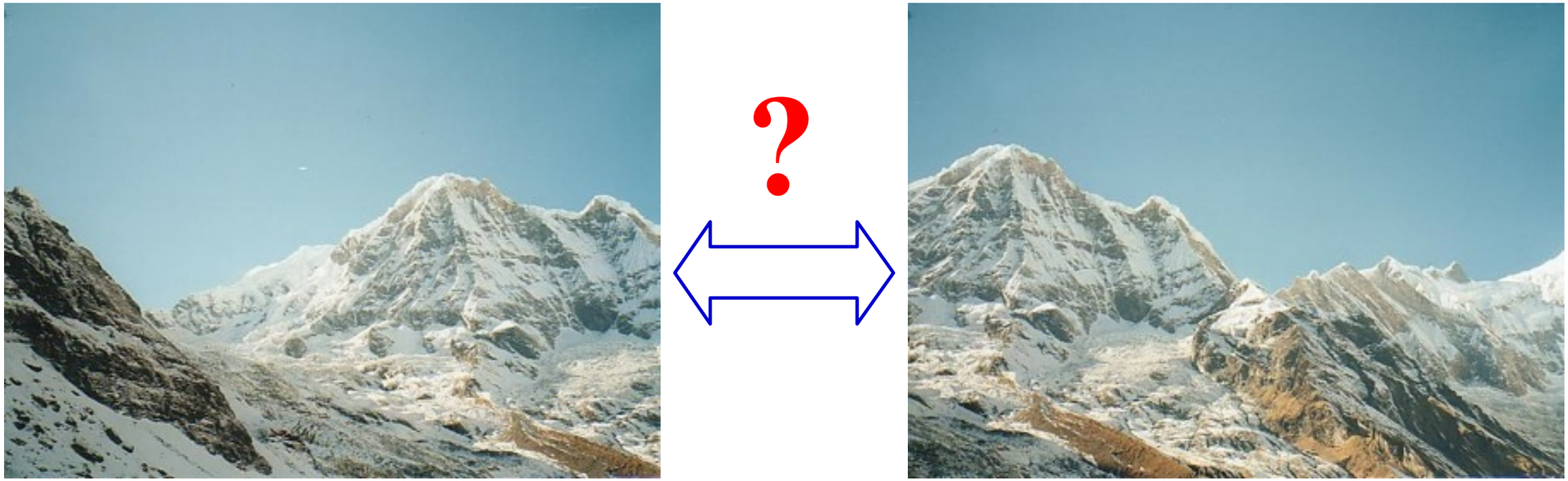
Application: Panorama stitching



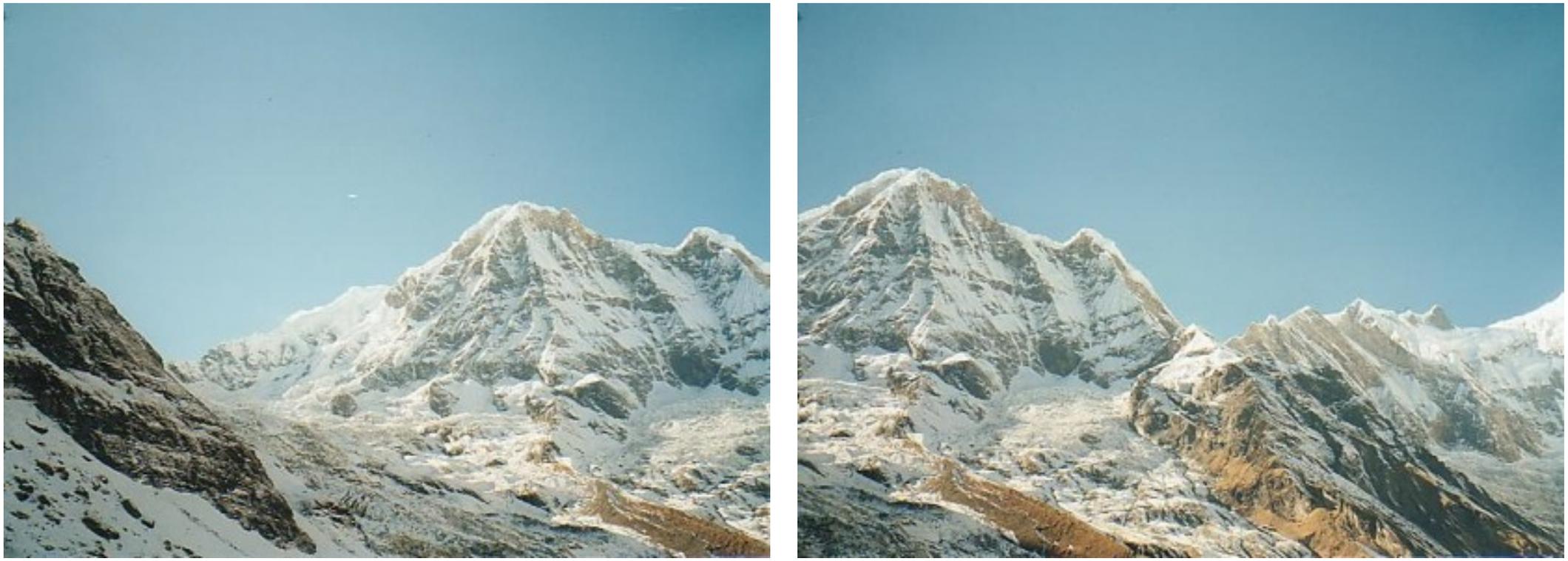
Source: Hartley & Zisserman

Panoramic stitching

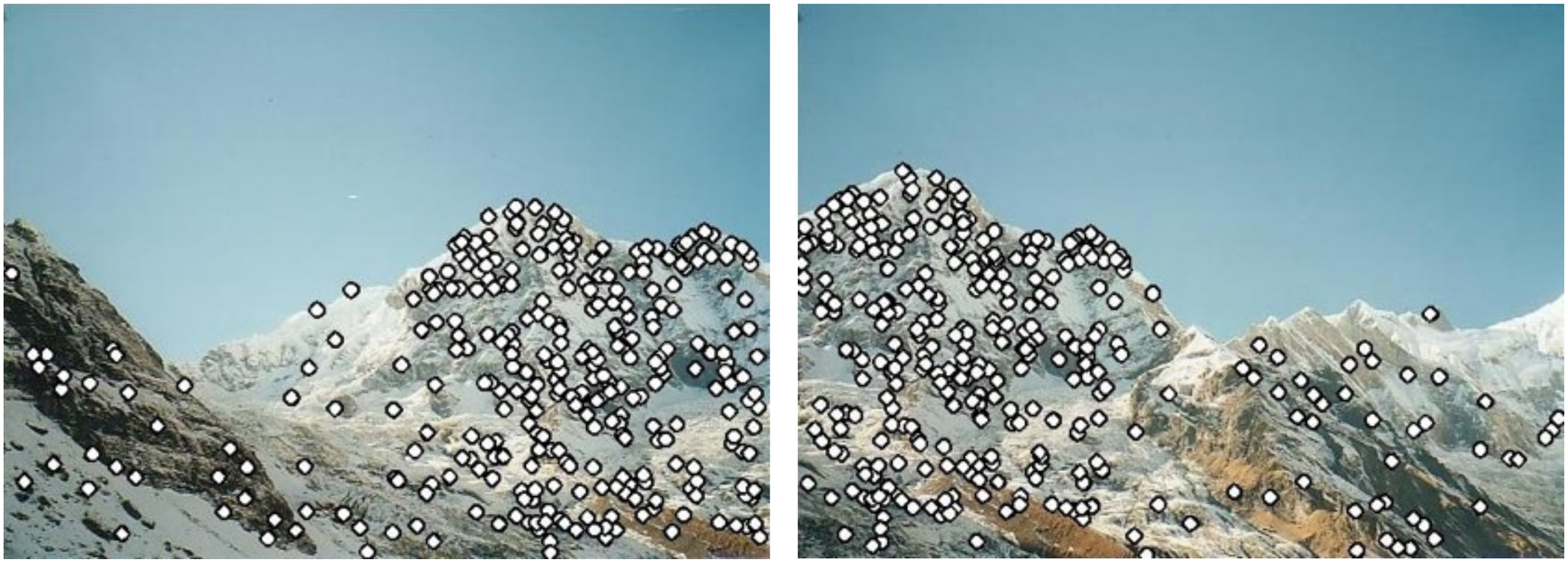
- **Approach**
  - Local feature matching
  - RANSAC for alignment



Panoramic stitching



Panoramic stitching

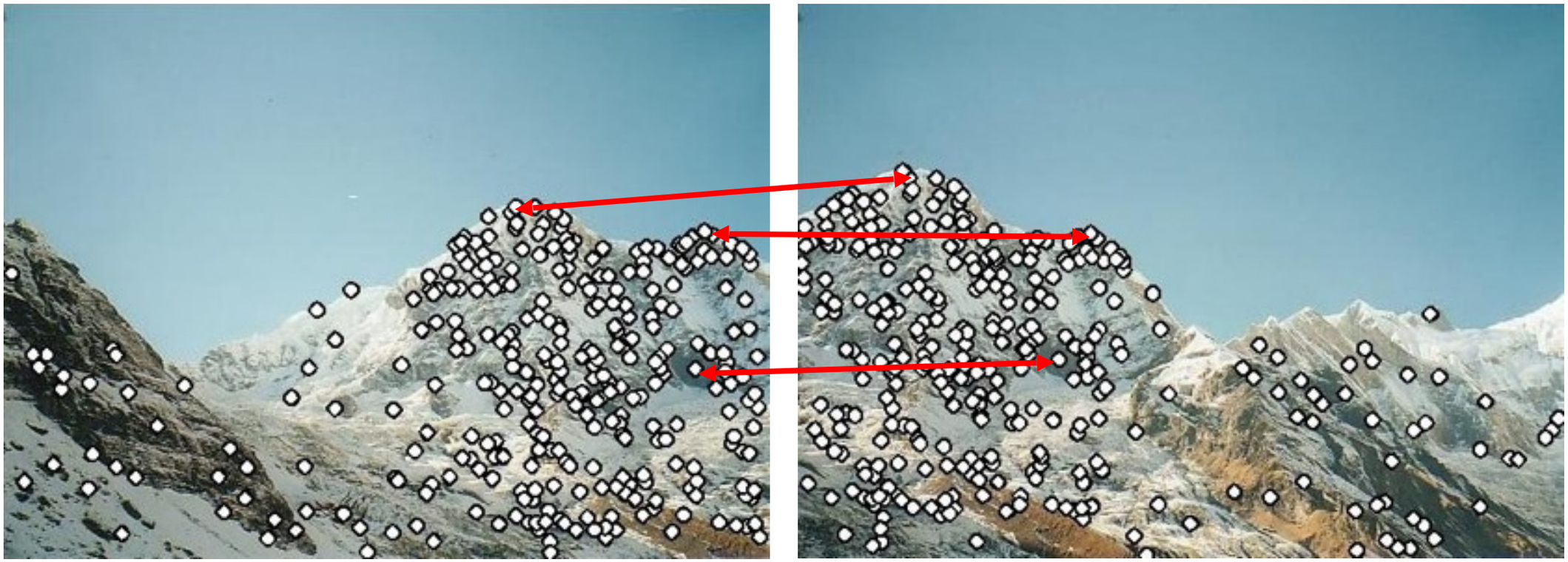


- Extract features
  - corner detector

Panoramic stitching

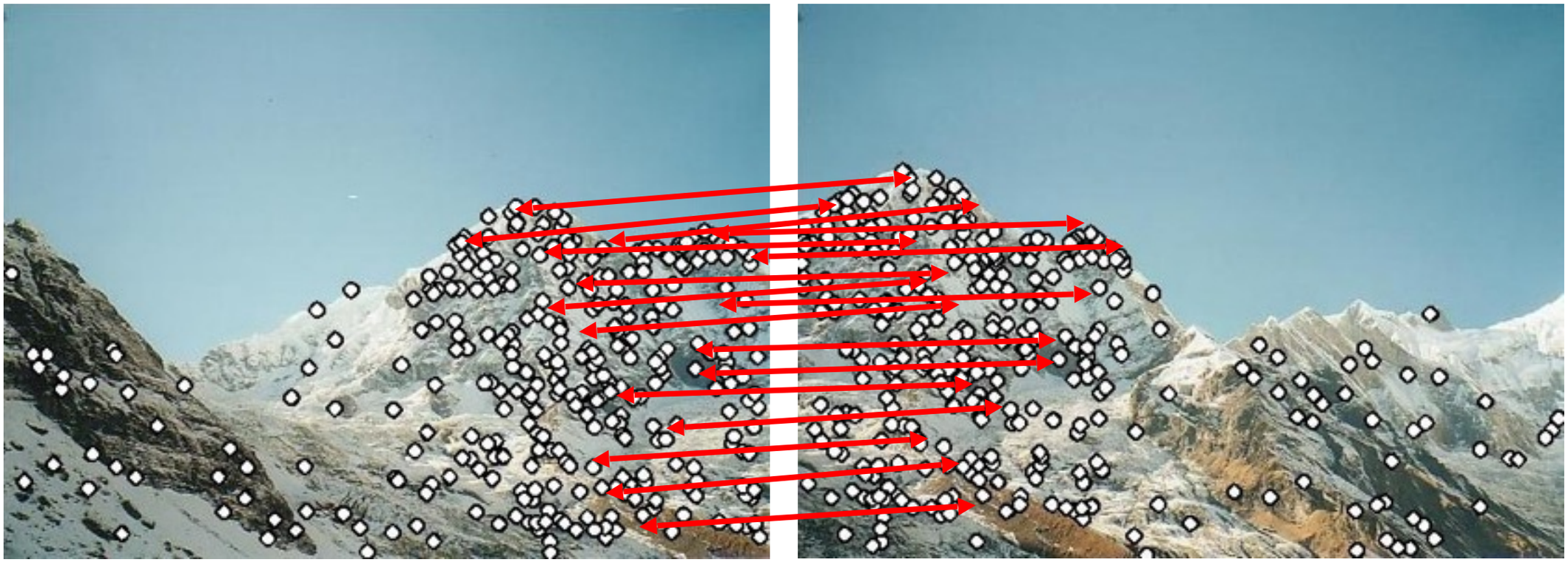
- Extract features
- Compute *putative matches*

Panoramic stitching



- Extract features
- Compute *putative matches*
- Loop:
  - *Hypothesize* transformation *T*

Panoramic stitching



- Extract features
- Compute *putative matches*
- Loop:
  - *Hypothesize* transformation *T*
  - *Verify* transformation (search for other matches consistent with *T*)

Panoramic stitching

- Extract features
- Compute *putative matches*
- Loop:
  - *Hypothesize* transformation *T*
  - *Verify* transformation (search for other matches consistent with *T*)

City-scale alignment

Microsoft Photosynth

https://www.youtube.com/watch?v=y9zF97JL30A

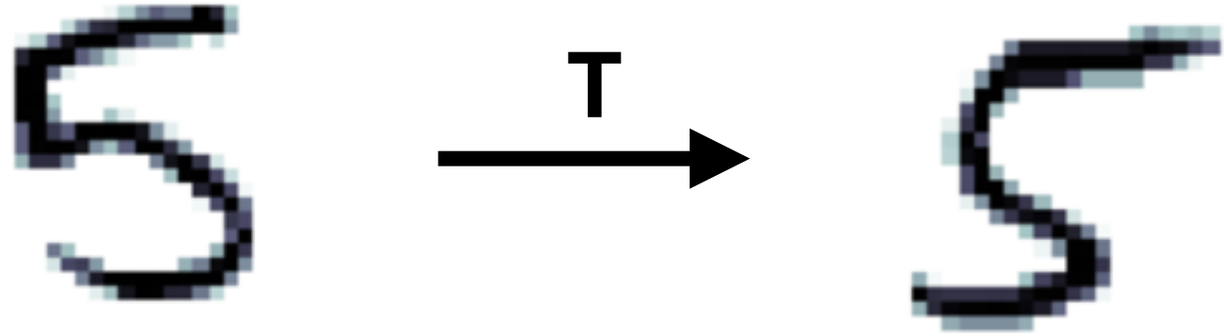
Alignment

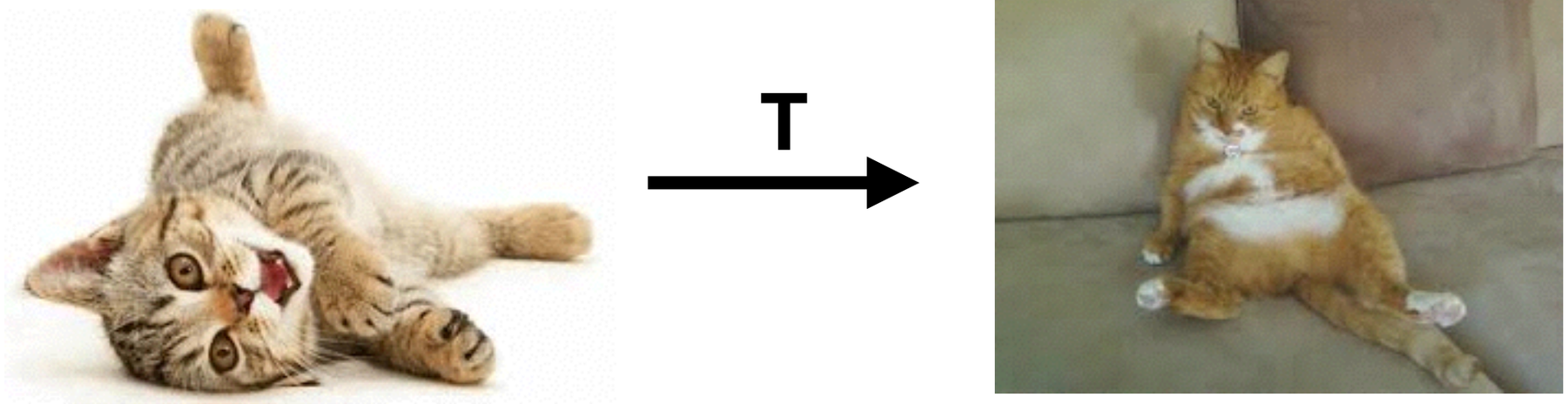
- What are the alignment problems in computer vision?
  - rigid vs. deformable (non-rigid)

- Good features to match
  - Invariance properties
  - Local features: the SIFT descriptor

- Alignment algorithms
  - Rigid alignment: RANSAC
    - Application: panoramic photo stitching
  - Non-rigid alignment
    - Application: shape matching

Non-rigid transformations





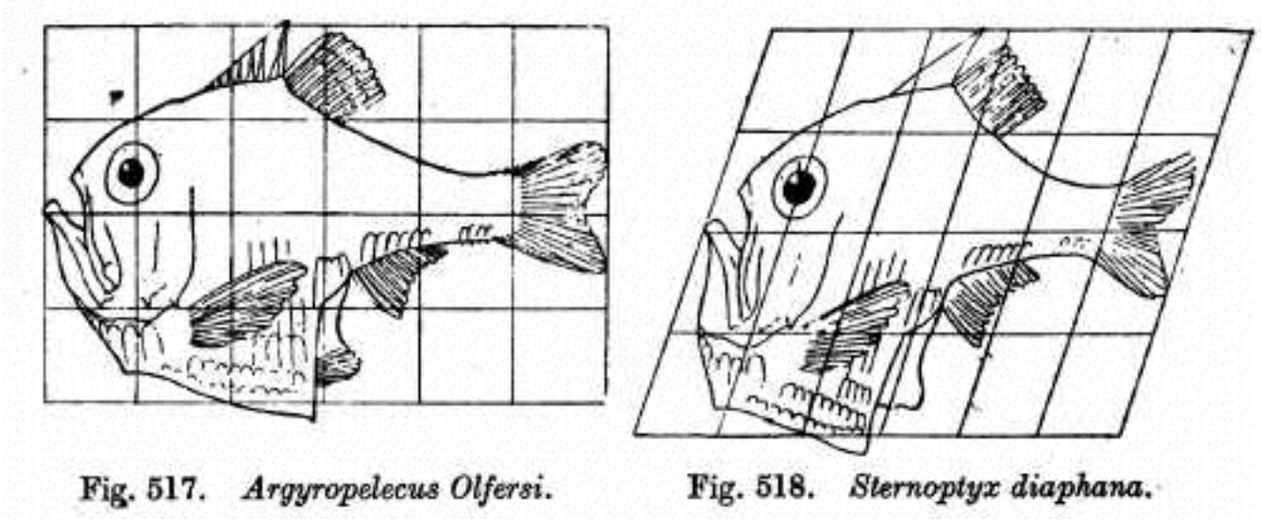
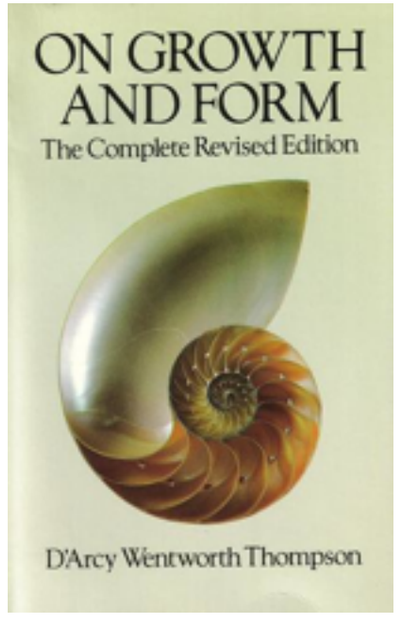


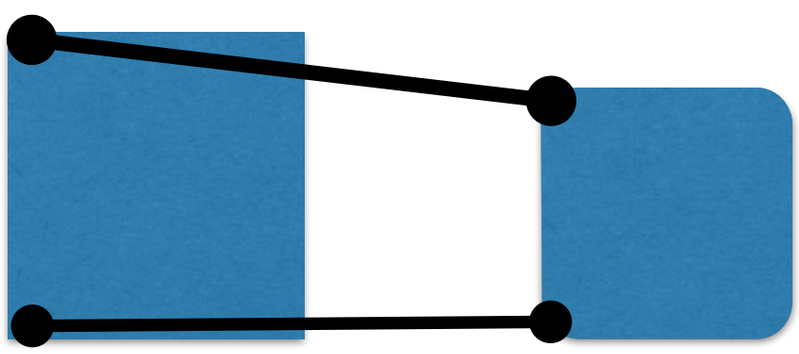
Fig. 517. *Argyropelecus Olfersi.* Fig. 518. *Sternoptyx diaphana.*

On Growth and Form, D'Arcy Thompson 1915

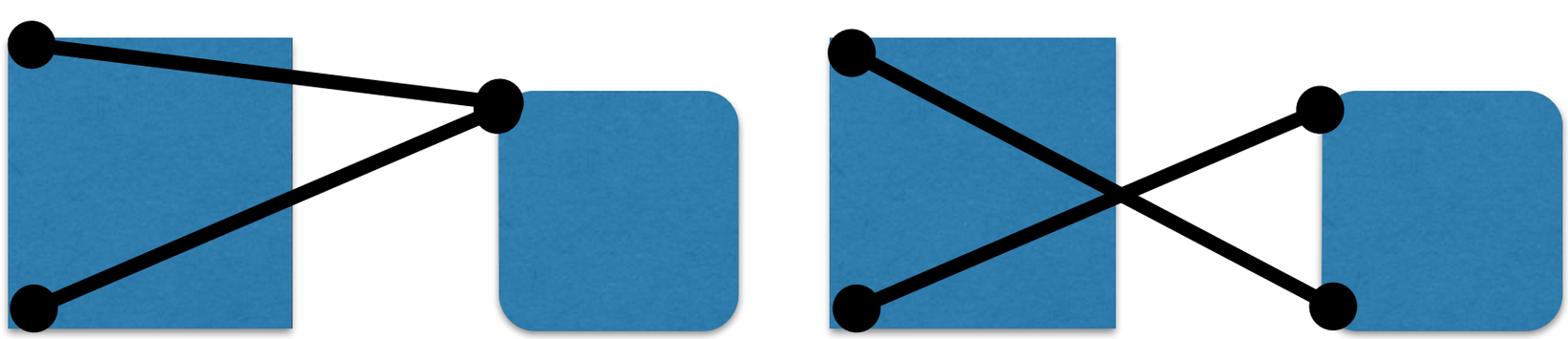


Non-rigid transformations

- How to characterize a non-rigid transformation?
- Global constraints
  - Should be from a family of constraints, e.g. Affine
- Local constraints
  - Low distortion, e.g., a good match should avoid
    - Many to one matches, one to many matches, criss crossings

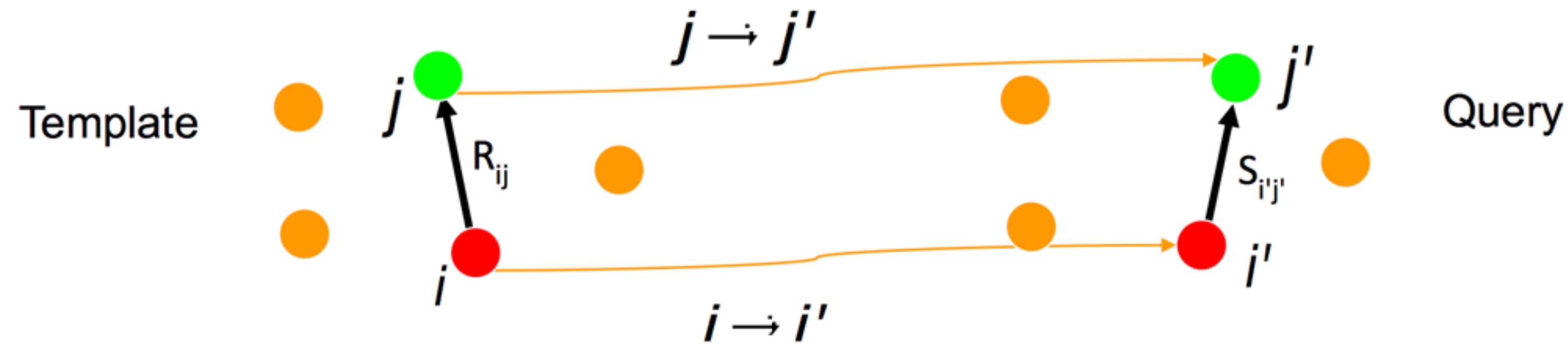


low distortion



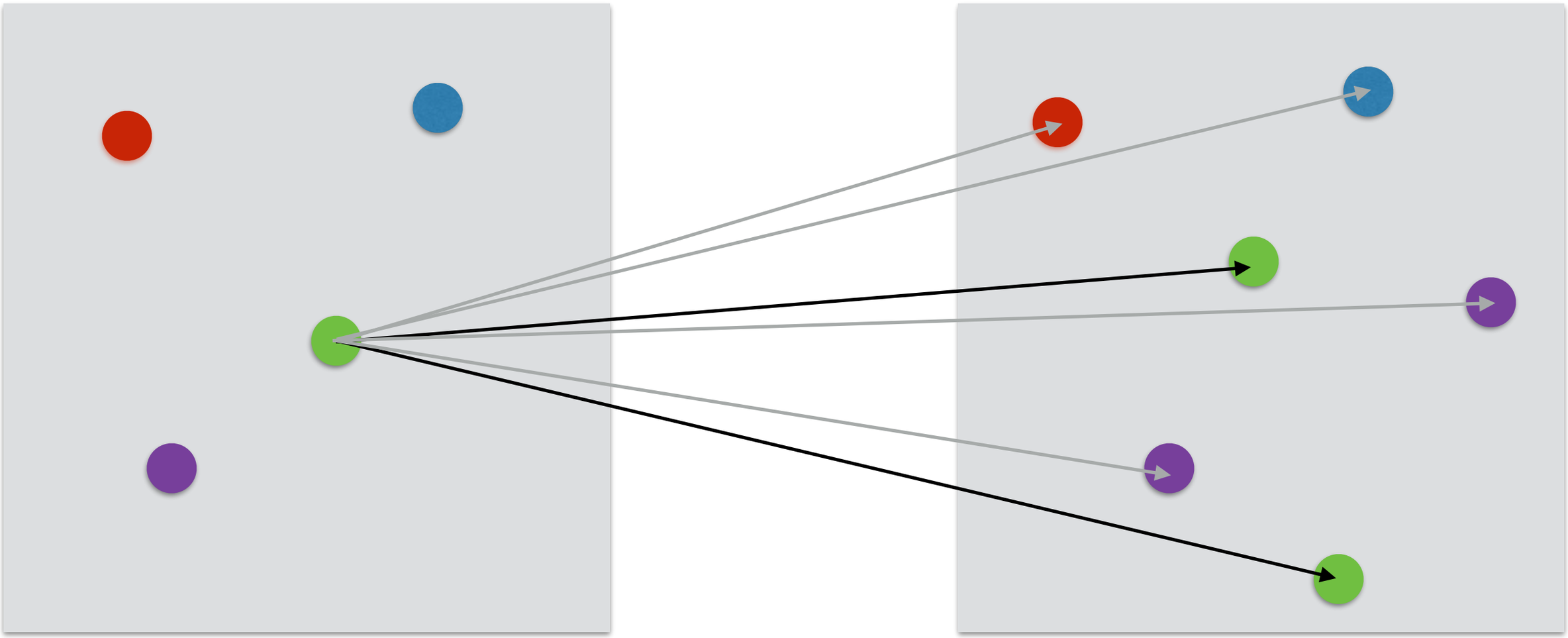
high distortion

Low distortion matching



- Measuring distortion between vectors
  - **R** and **S** same *length* for *rotations*
  - **R** and **S** same *angle* for *scaling*
- Distortion cost: $\alpha f_r\left(|R_{ij}|, |S_{i'j'}|\right) + (1-\alpha) f_s\left(\Theta(R_{ij}), \Theta(S_{i'j'})\right)$
- Matching score: $c(i \rightarrow i') + c(j \rightarrow j')$

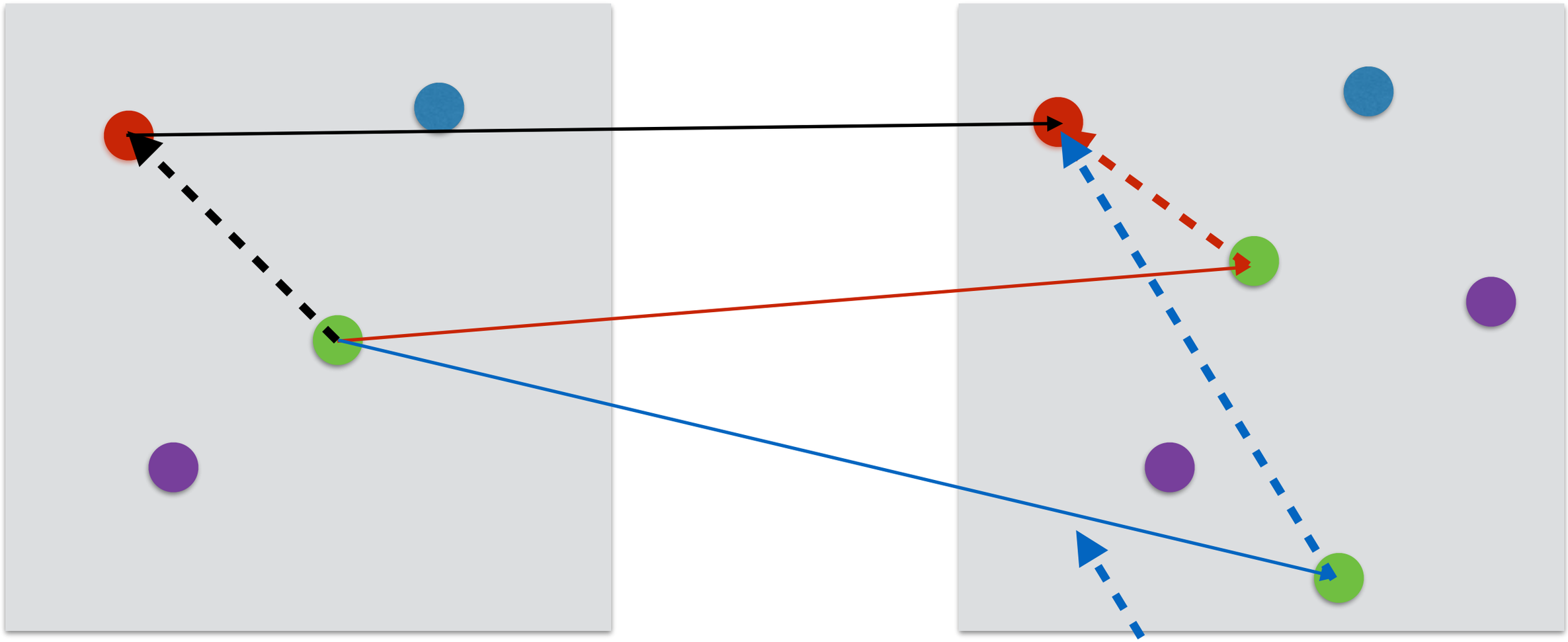
Scoring matches



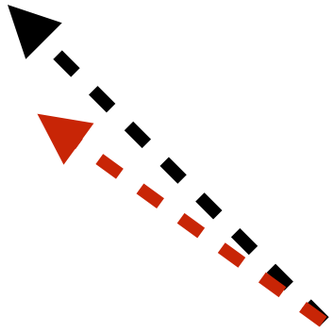
possible matches for the green point

— bad matches — good matches

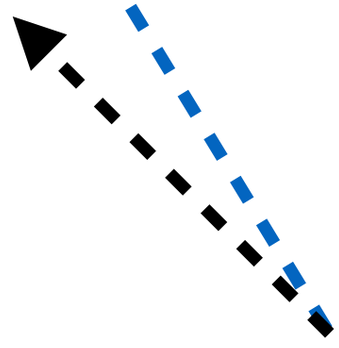
Scoring matches



fix the red match

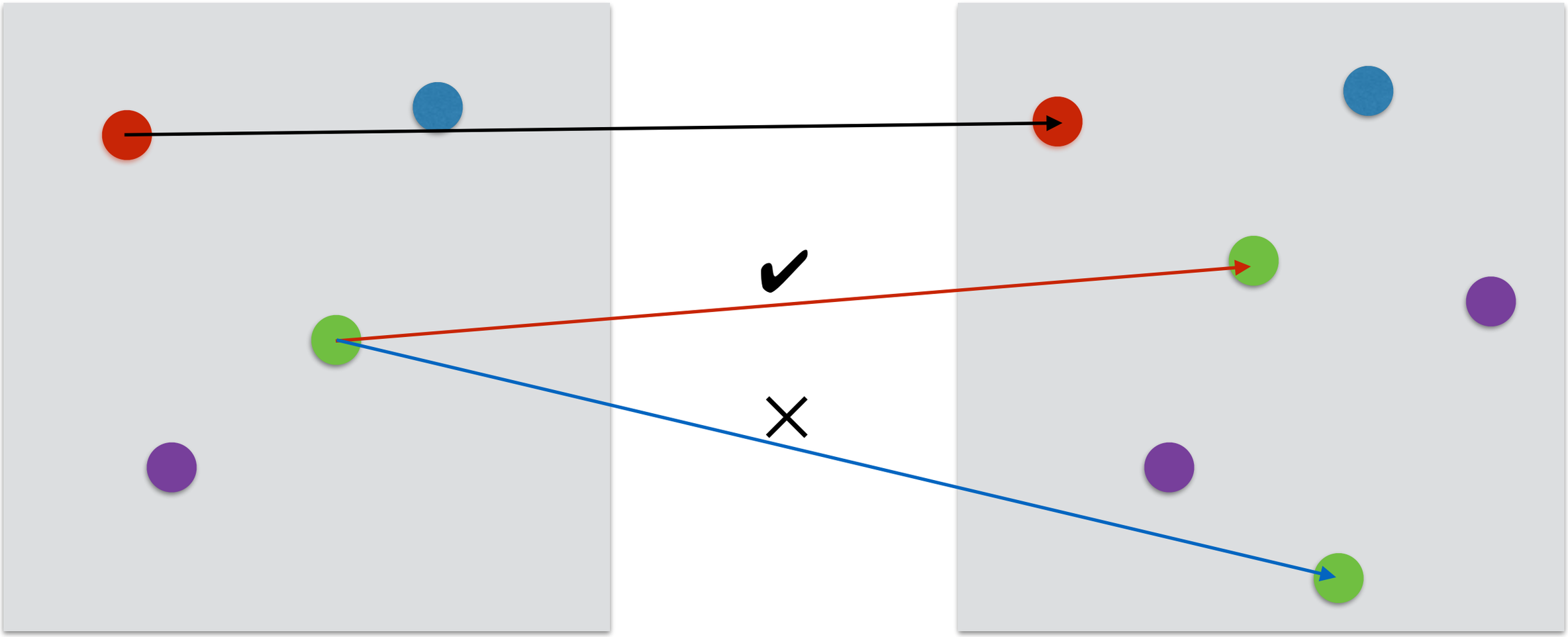


low distortion



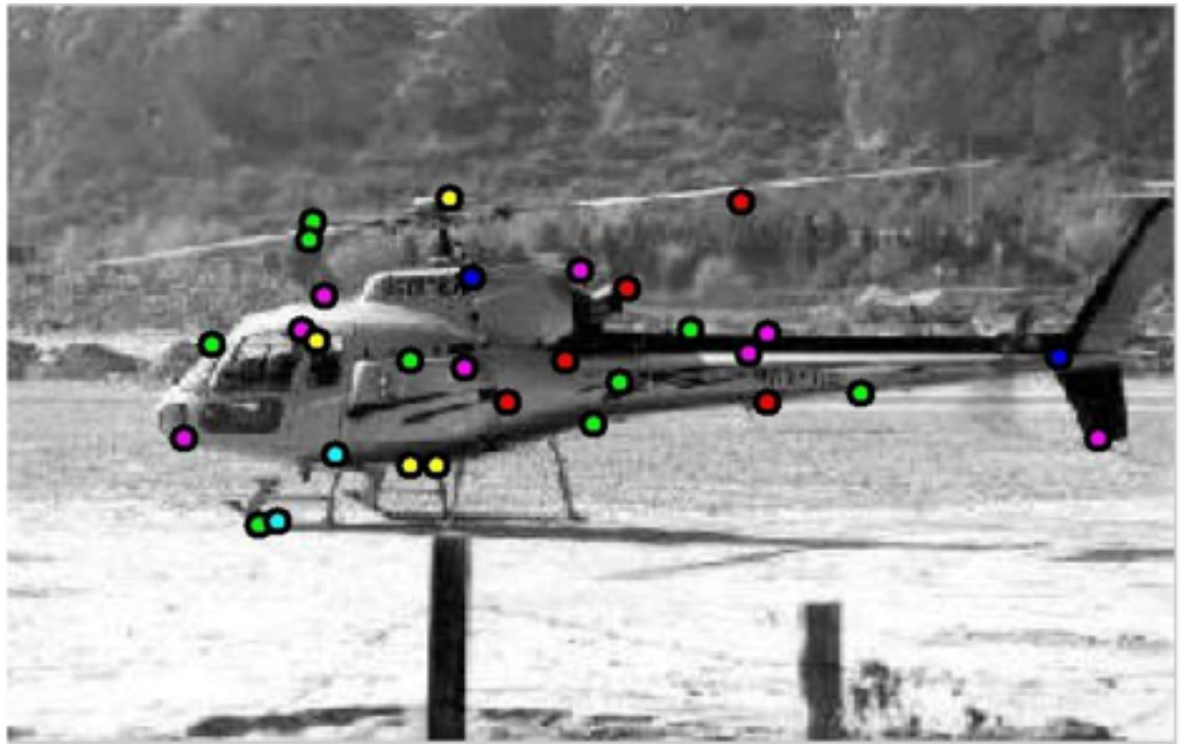
high distortion

Scoring matches



fix the red match

Correspondence algorithm



- Find correspondence that:
  - minimizes distortion (pairwise)
  - maximizes scores (individual) based on local descriptors matching

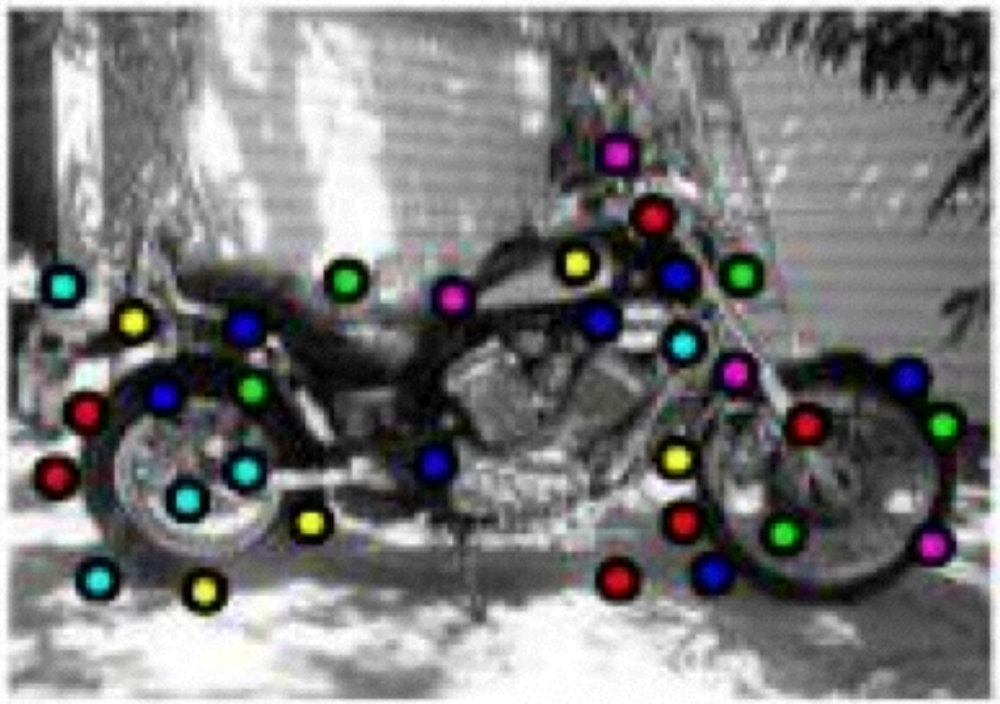
$$minimize\left(-\sum_{a} score(a) + \sum_{a,b} distortion(a,b)\right)$$

Correspondence algorithm

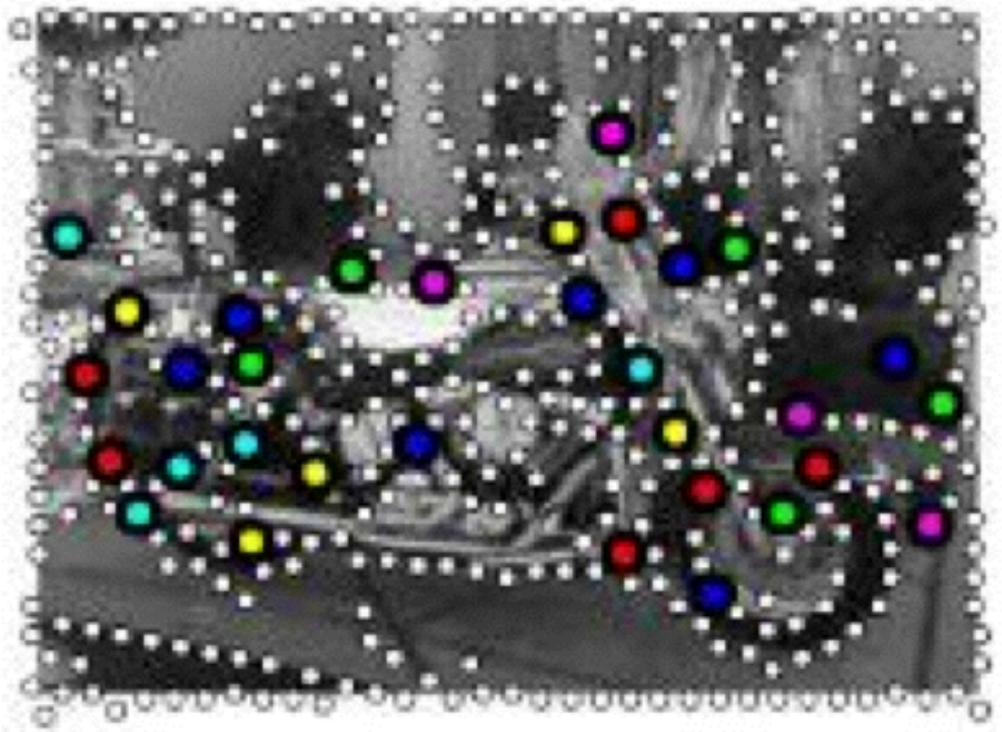
- Find correspondence that:
  - minimizes distortion (pairwise)
  - maximizes scores (individual) based on local descriptors matching
- Objective is an integer quadratic program (IQP)
  - NP-Hard
  - But approximate solutions can be found in reasonable time

Example matches

Probe

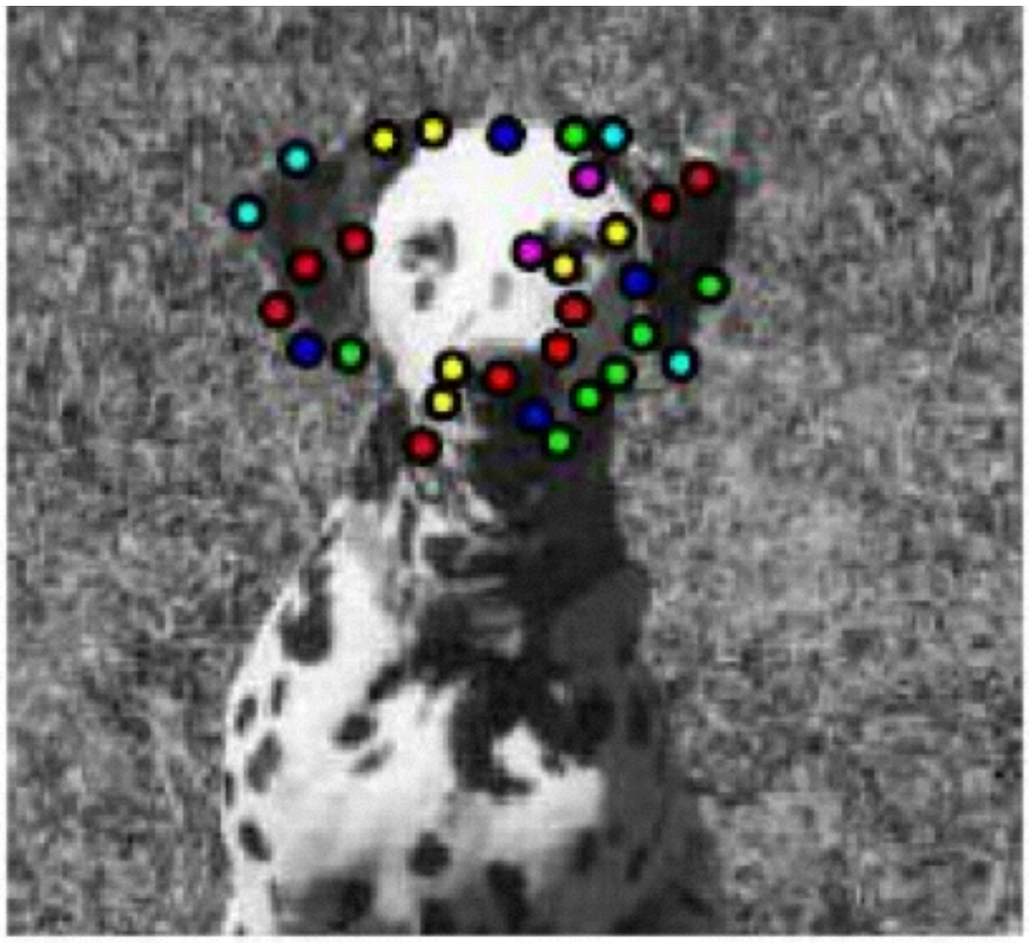


Query

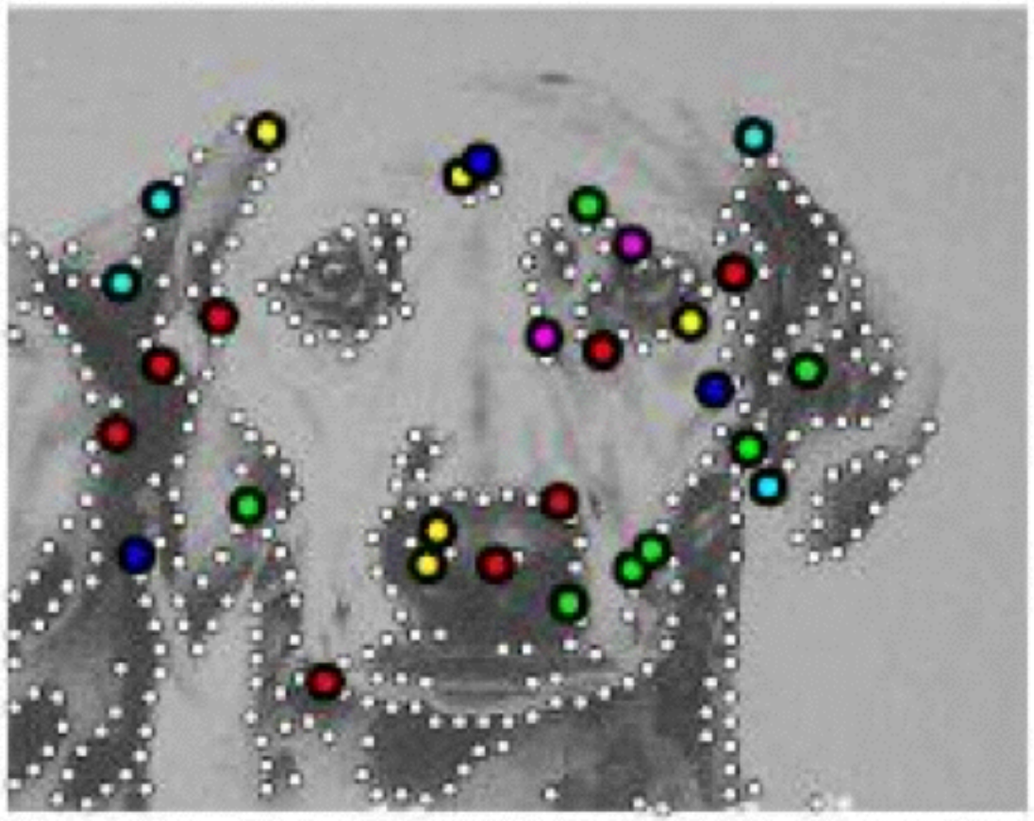


Example matches

Probe



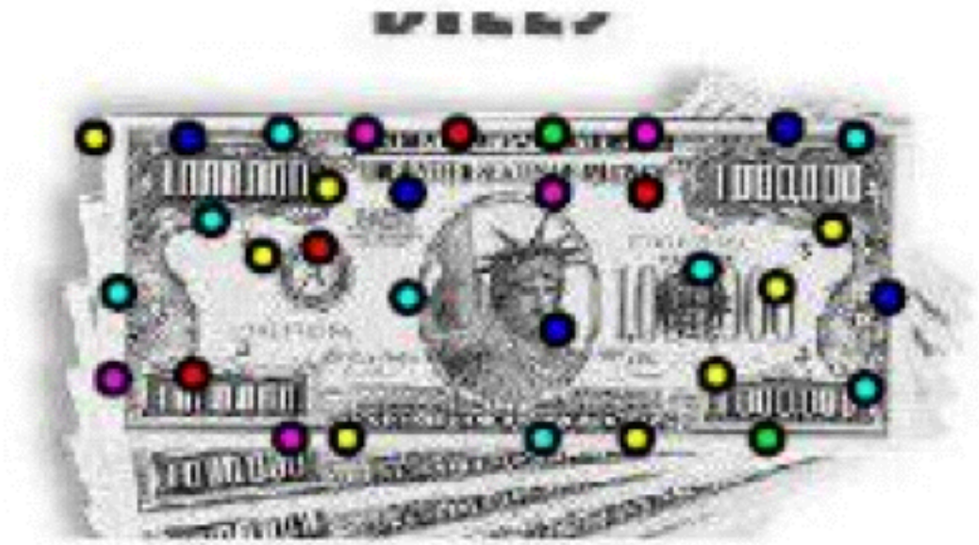
Query



Berg et al., CVPR 2005

Example matches

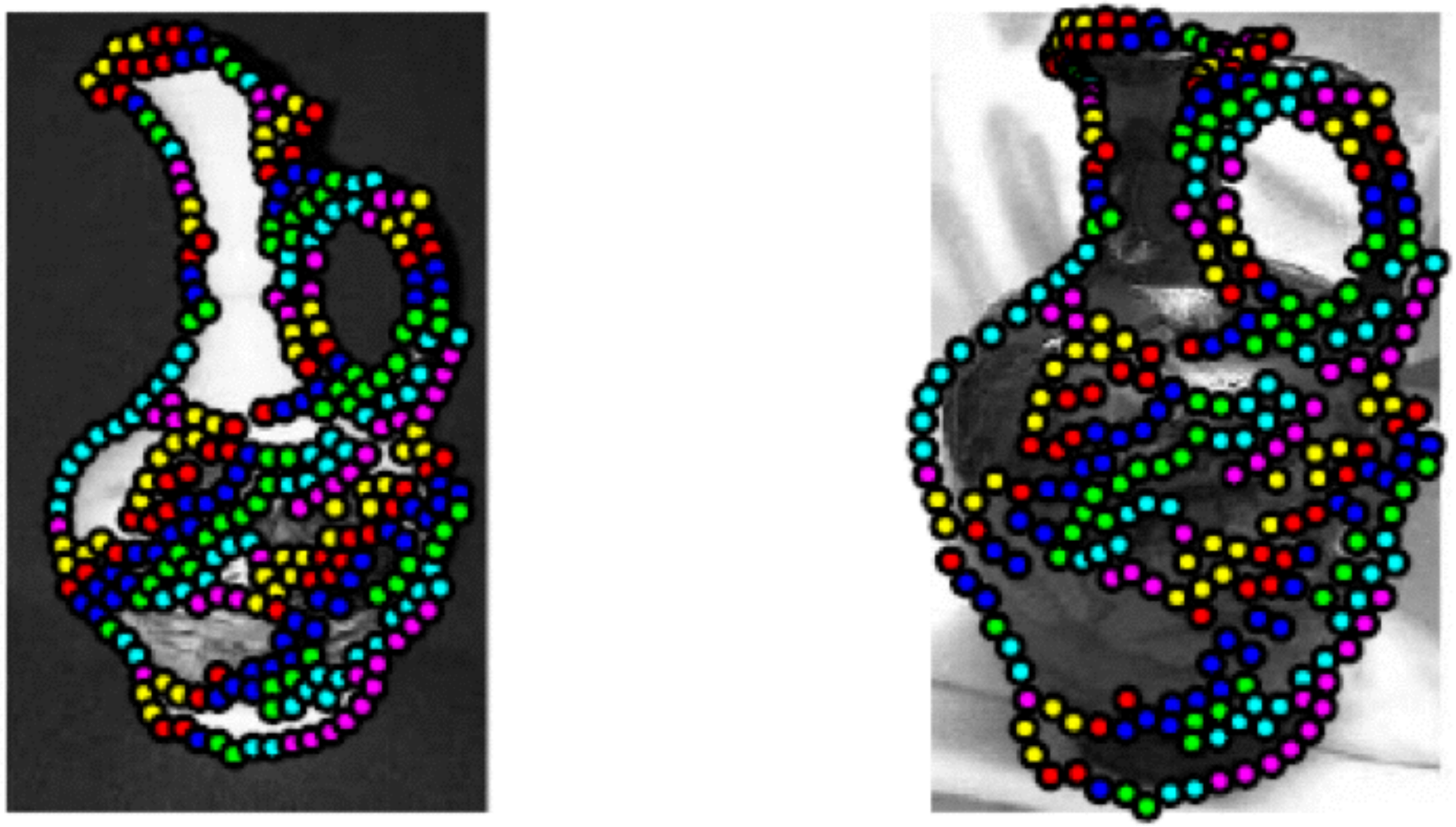
Probe



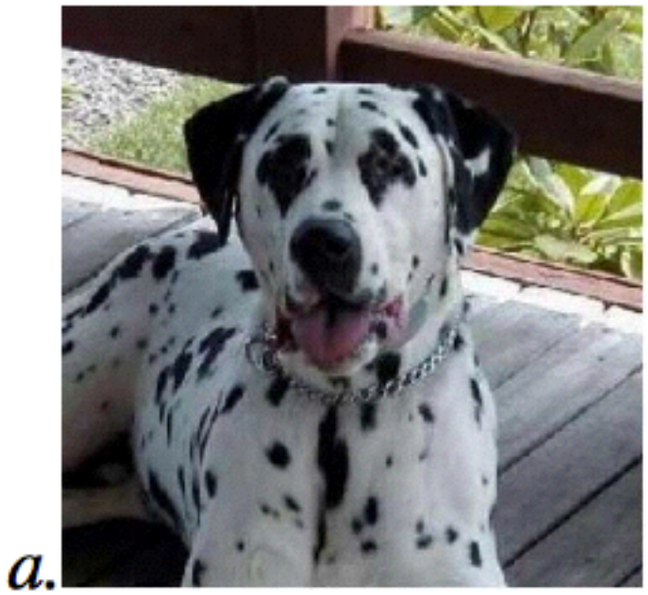
Query

Example matches

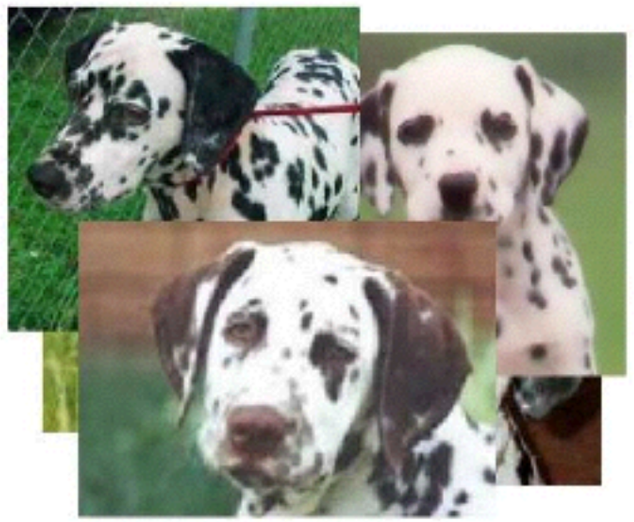
Interpolating the alignment



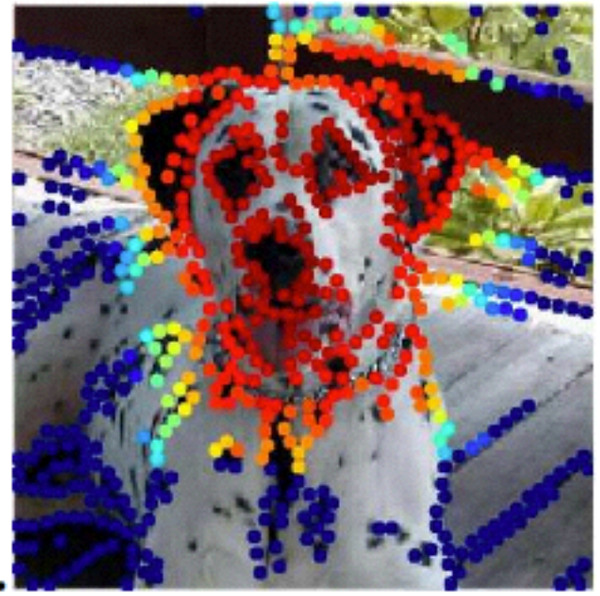
Automatic segmentation



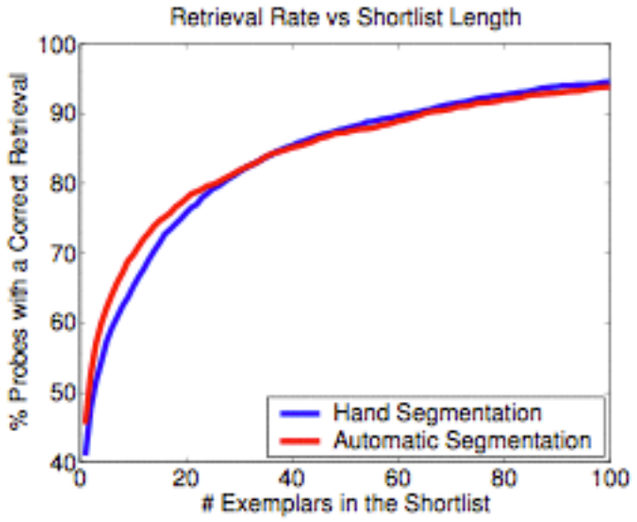
a.



b.



c.



d.

Berg et al., CVPR 2005

Alignment

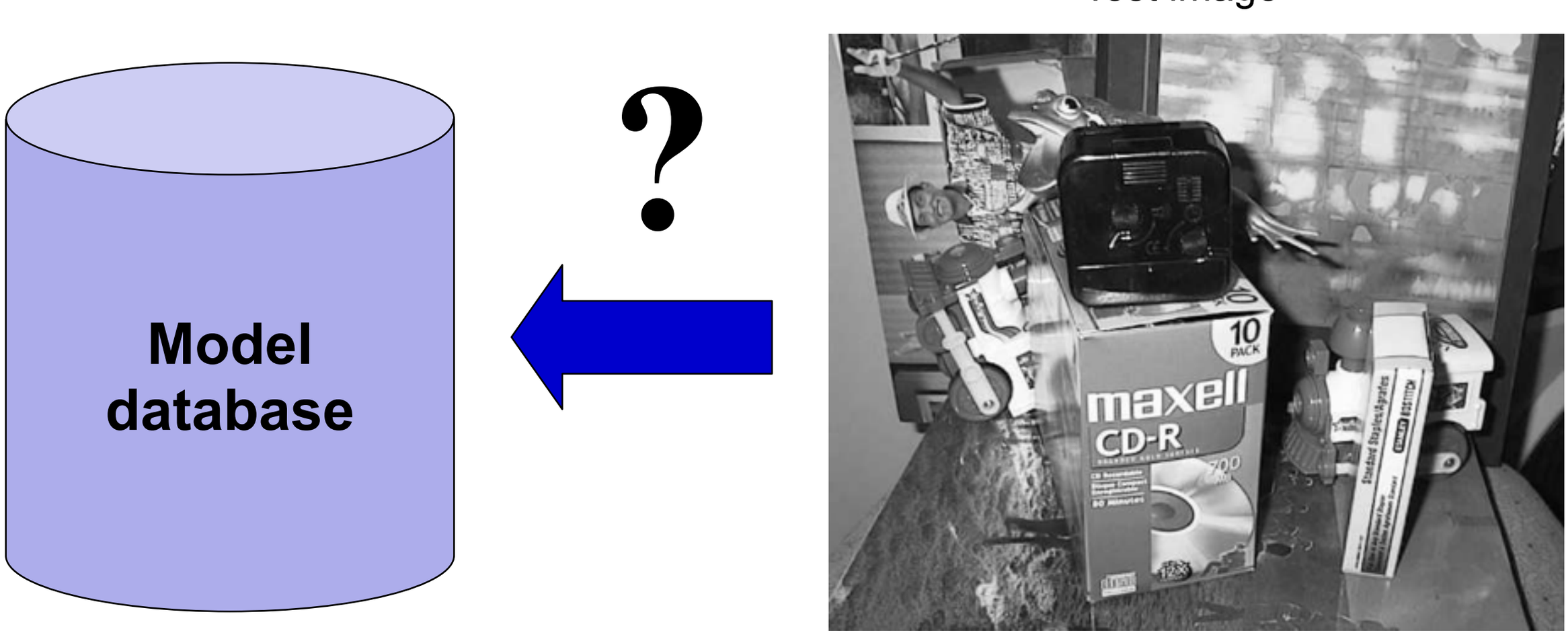
- What are the alignment problems in computer vision?
  - rigid vs. deformable (non-rigid)

- Good features to match
  - Invariance properties
  - Local features: the SIFT descriptor

- Alignment algorithms
  - Rigid alignment: RANSAC
    - Application: panoramic photo stitching
  - Non-rigid alignment
    - Application: shape matching

Scalability: Alignment to large databases

- What if we need to align a test image with thousands or millions of images in a model database?
  - Efficient putative match generation
    - Approximate descriptor similarity search, inverted indices



Large-scale visual search

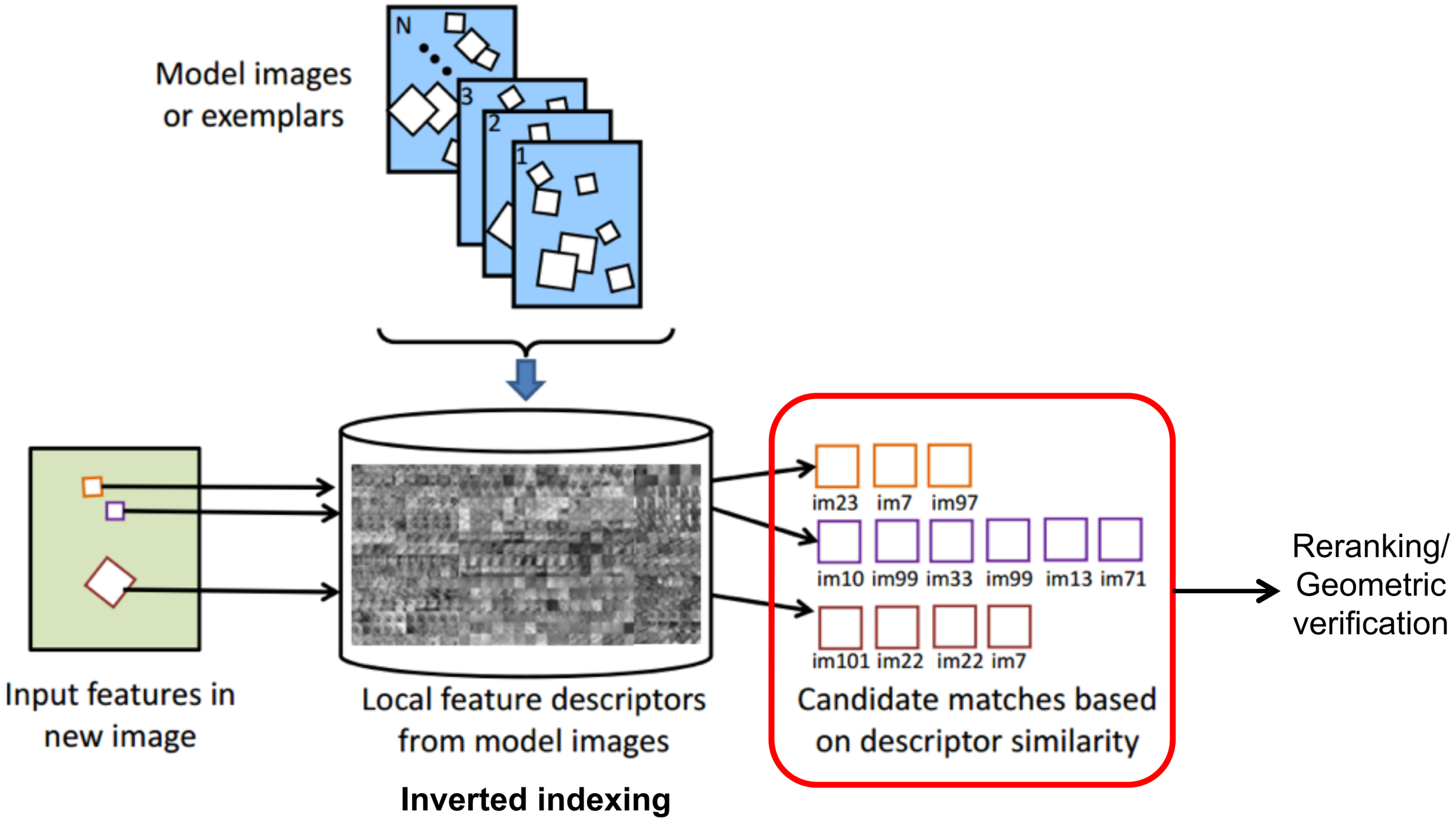
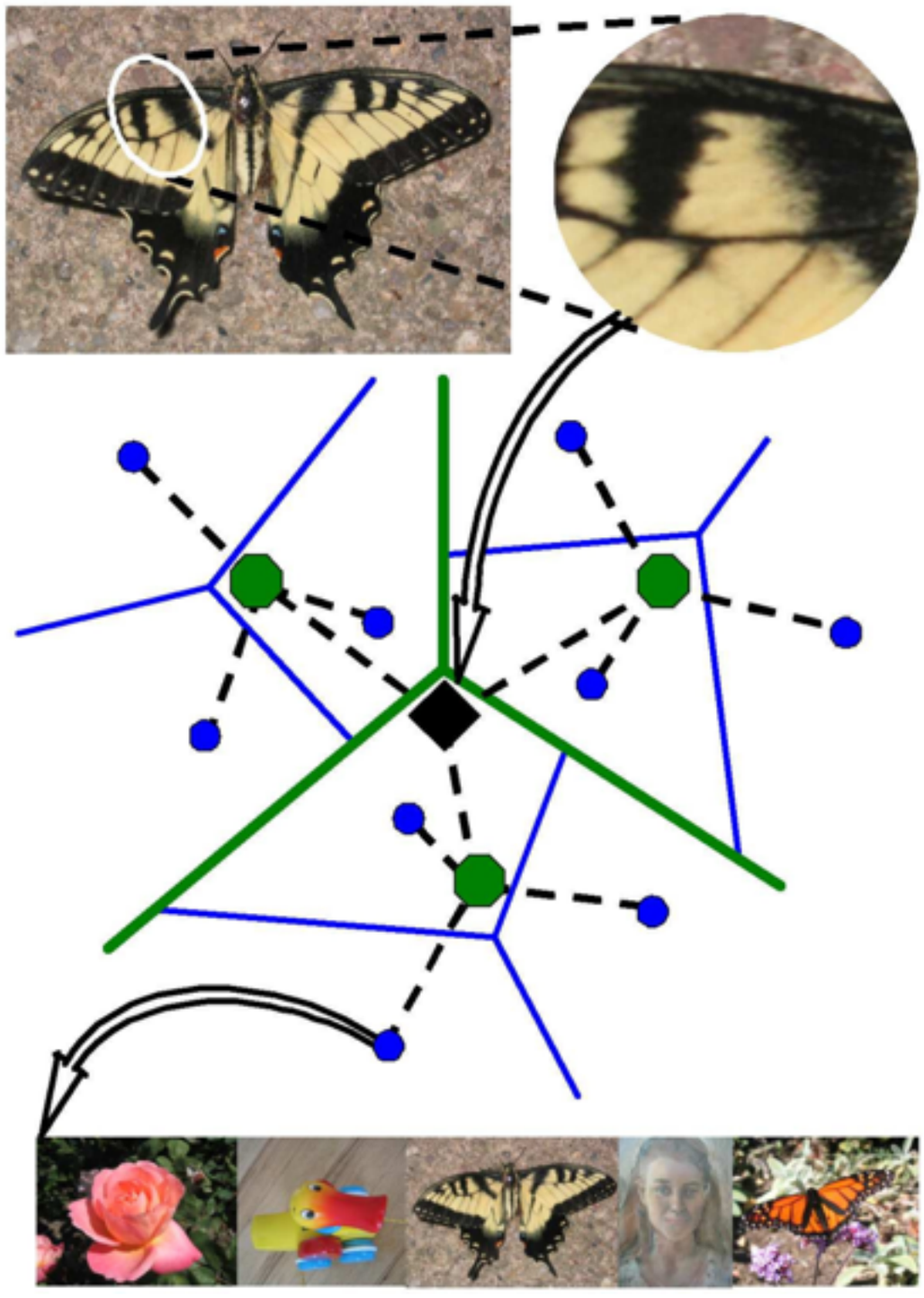


Figure from: Kristen Grauman and Bastian Leibe, Visual Object Recognition, Synthesis Lectures on Artificial Intelligence and Machine Learning, April 2011, Vol. 5, No. 2, Pages 1-181

Example indexing technique: Vocabulary trees

Test image



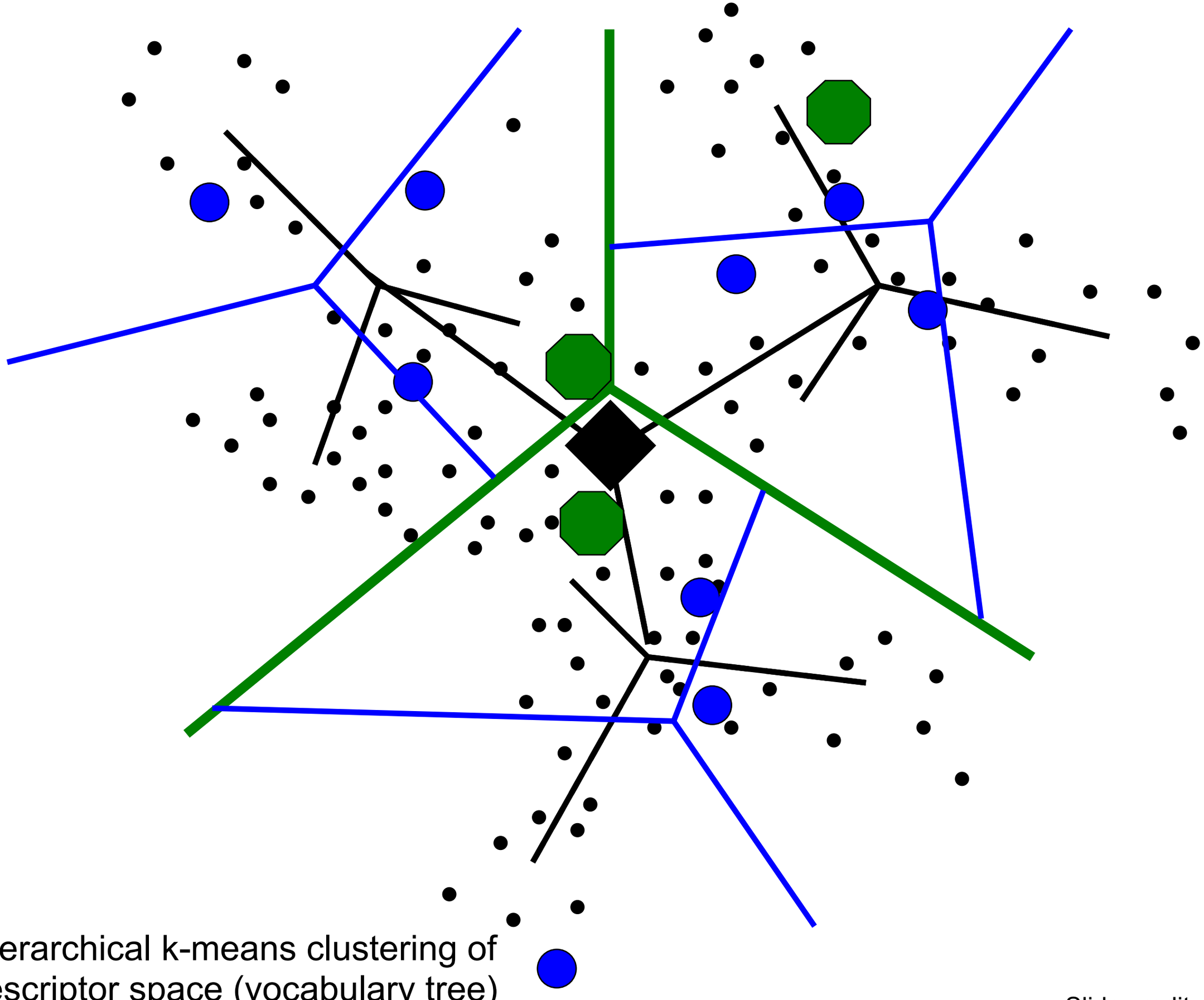
Vocabulary tree with inverted index

Database

D. Nistér and H. Stewénius, Scalable Recognition with a Vocabulary Tree, CVPR 2006

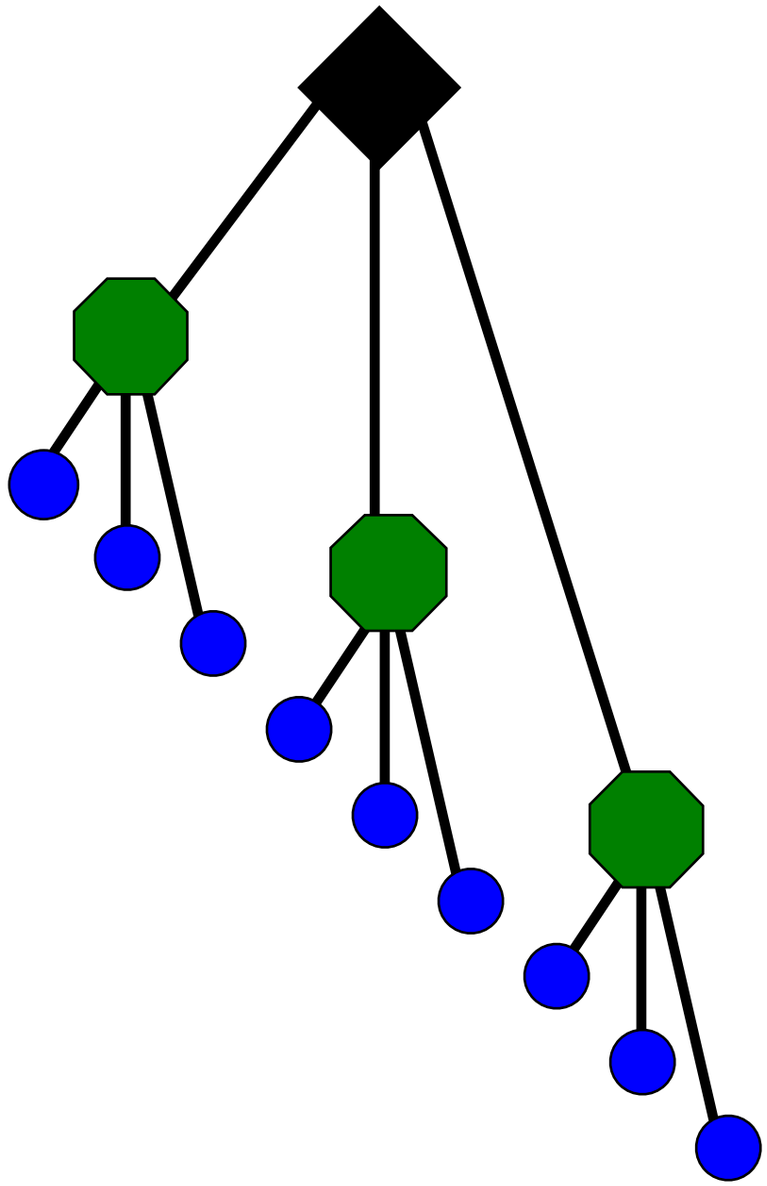
Descriptor space

Goal: find a set of representative prototypes or cluster centers to which descriptors can be quantized



Hierarchical k-means clustering of descriptor space (vocabulary tree)

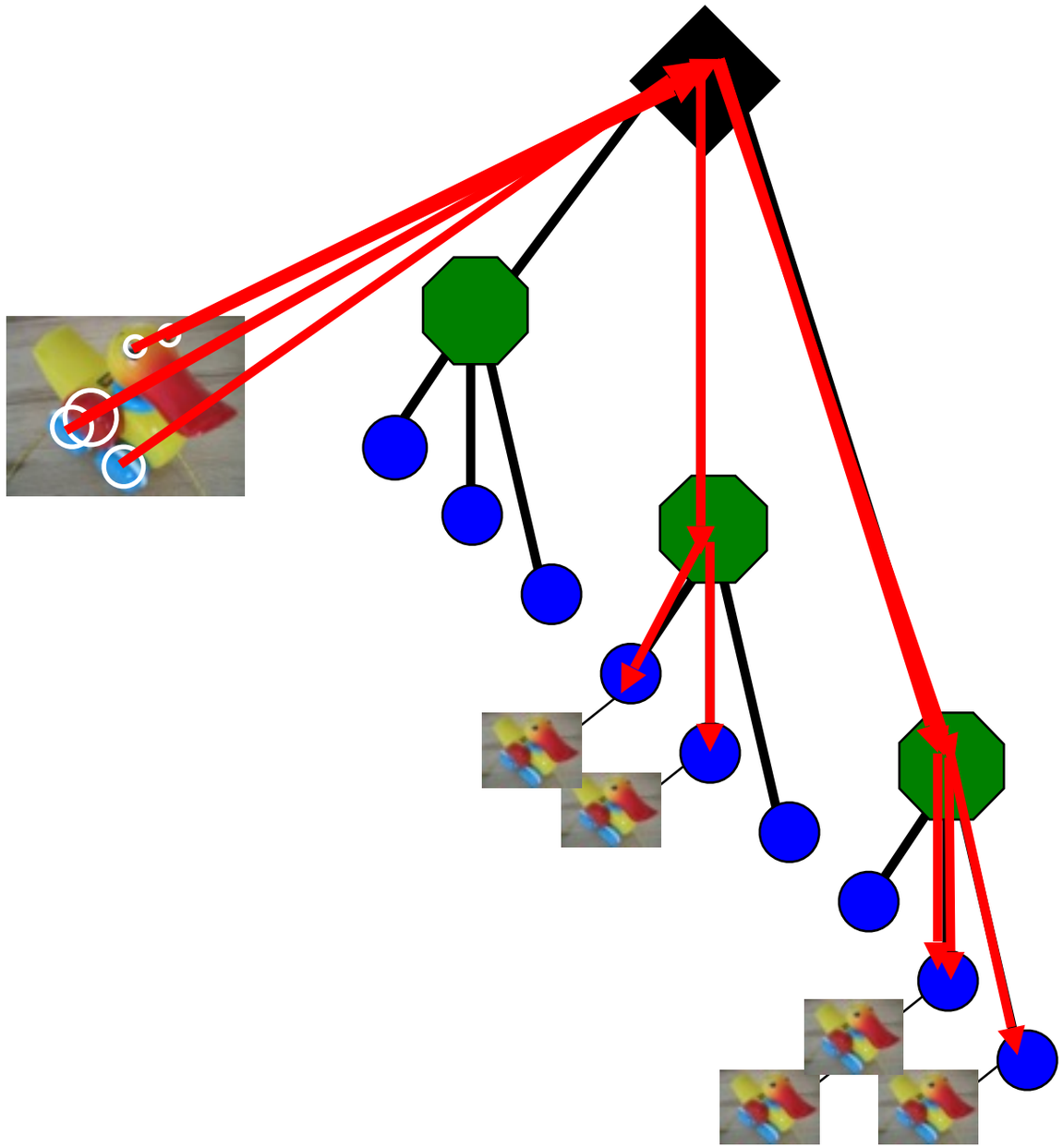
Slide credit: D. Nister



Vocabulary tree/inverted index

Slide credit: D. Nister

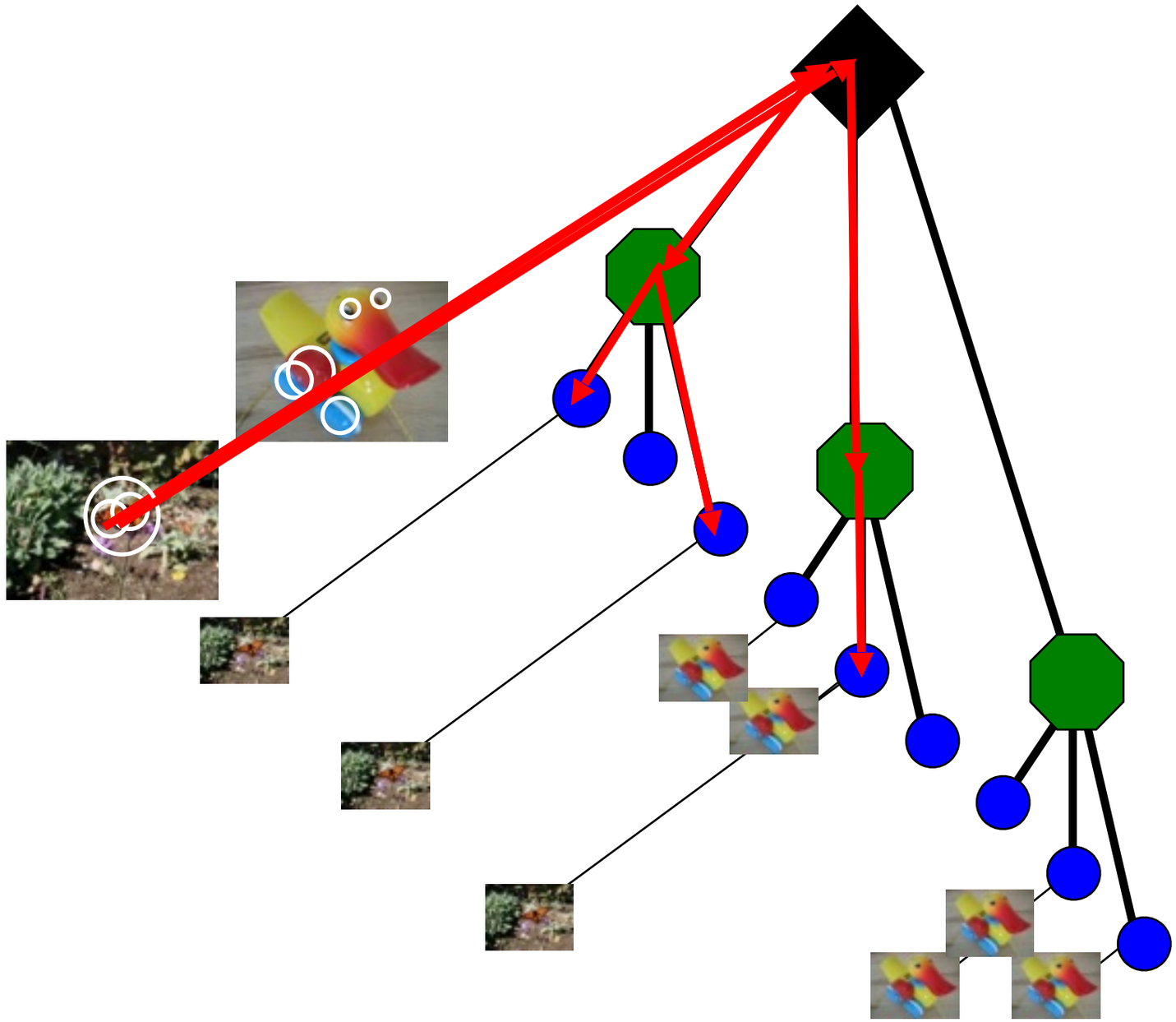
Model images



Populating the vocabulary tree/inverted index

Slide credit: D. Nister

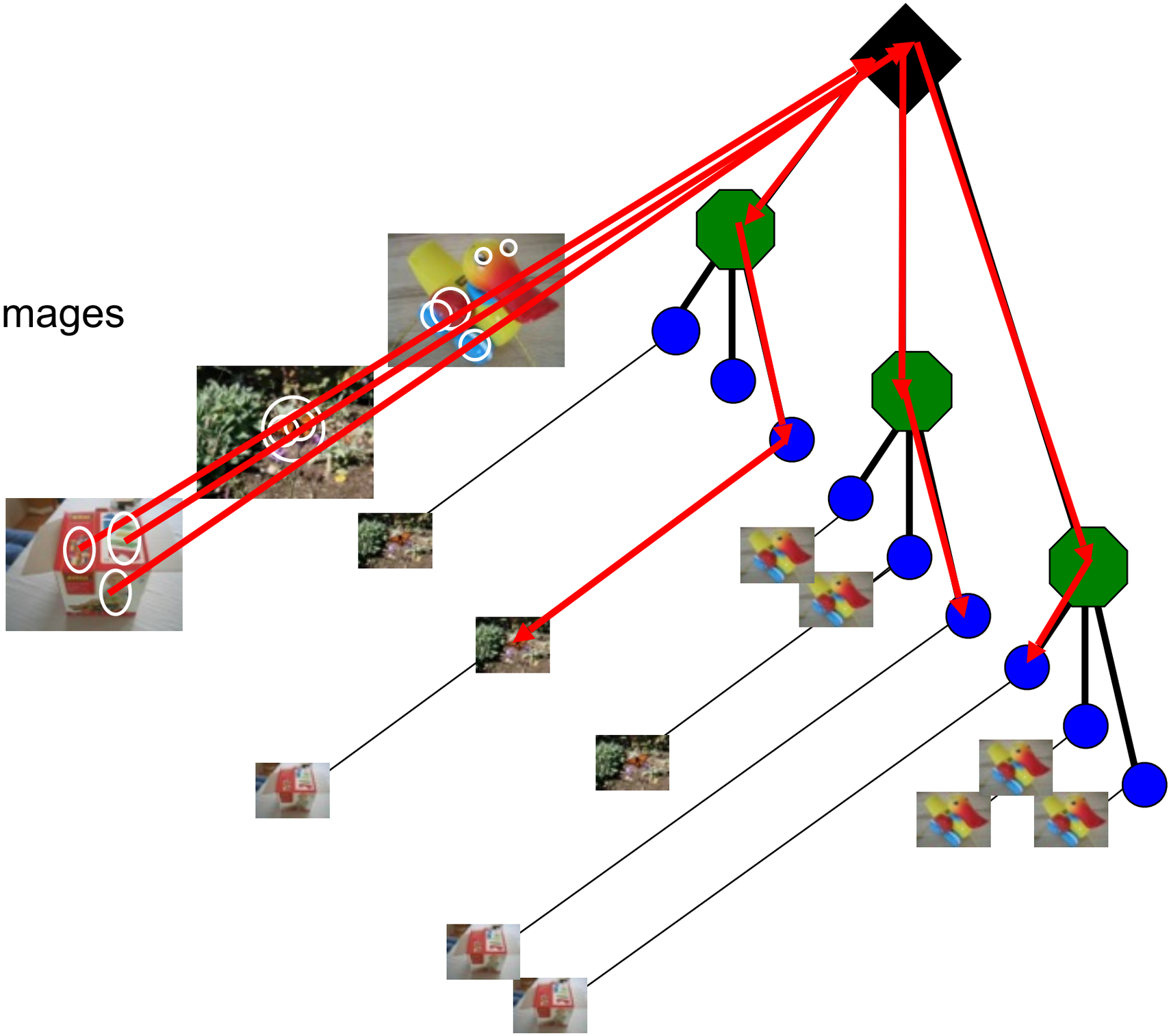
Model images



Populating the vocabulary tree/inverted index

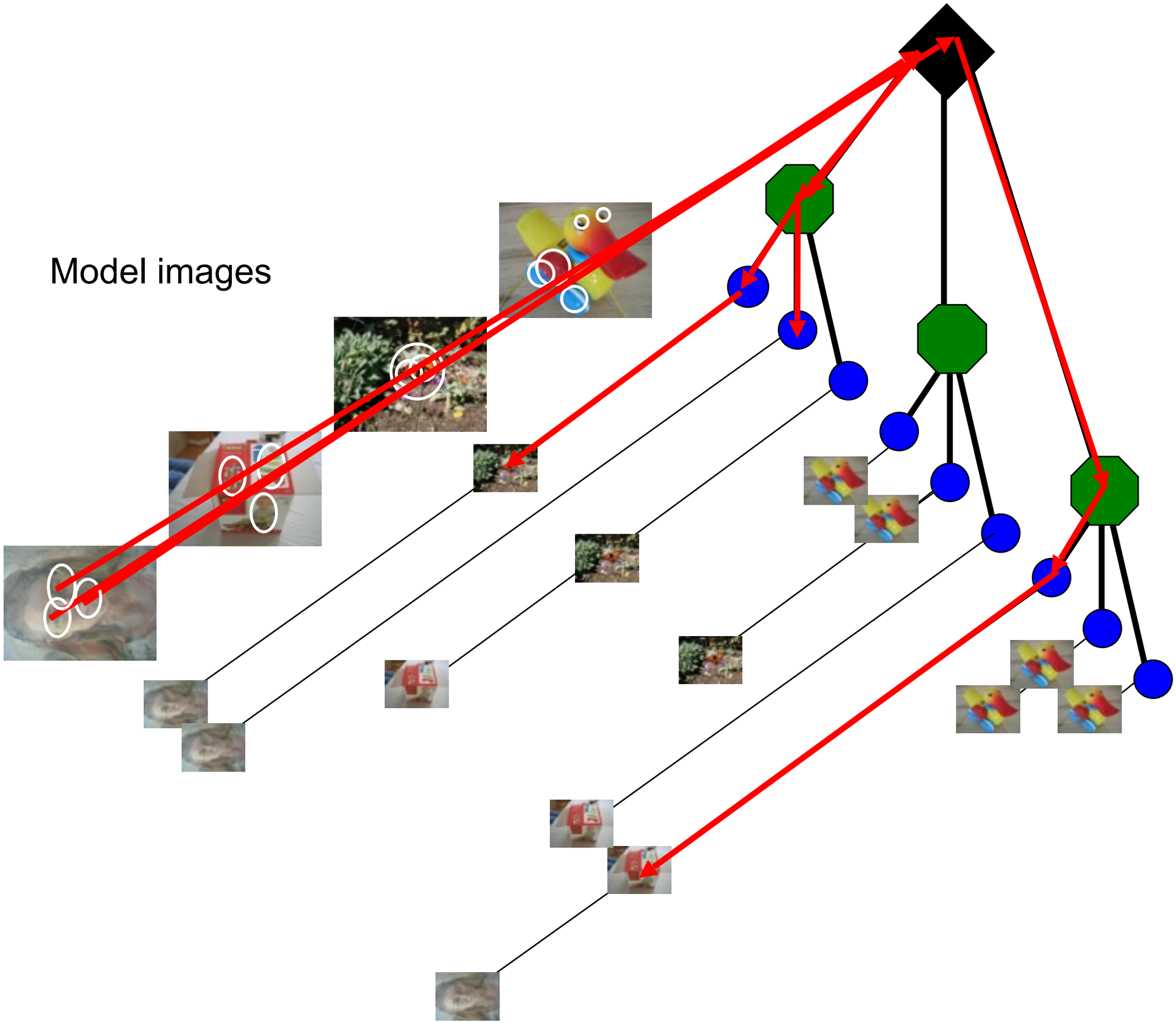
Slide credit: D. Nister

Model images



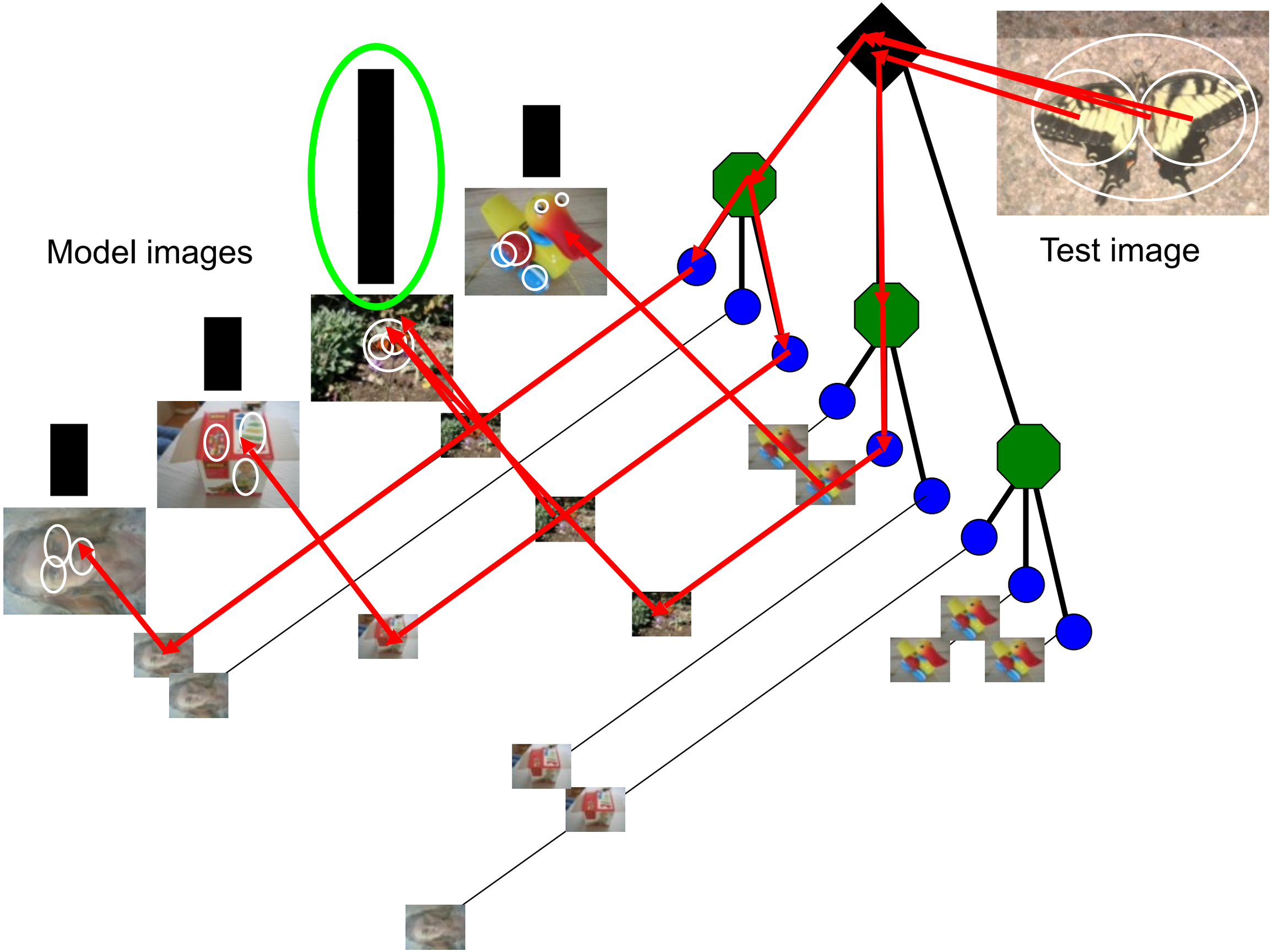
Populating the vocabulary tree/inverted index

Slide credit: D. Nister



Populating the vocabulary tree/inverted index

Slide credit: D. Nister



Looking up a test image

Slide credit: D. Nister

Further thoughts and readings …

- Chapters 6, 7 and 9 from Richard Szeliski's book.
- Shape matching references
  - Shape matching and object recognition using shape contexts, Belongie, Malik and Puzicha, PAMI 2002 (**paper**)
  - Shape matching and object recognition using low distortion correspondences, Berg, Berg and Malik, CVPR 2005 (**paper**)
  - Hierarchical matching of deformable shapes, Felzenszwalb and Schwartz, CVPR 2007 (**paper**)
  - David Nister's vocabulary tree **paper**
- Web demos from Oxford VGG group
  - **Video google**, **Oxford building search**, **Sculpture retrieval**

Cool application of large-scale alignment: searching the night sky

http://www.astrometry.net/